

# Enabling Process Mining on Sensor Data from Smart Products

van Eck, M.L.; Sidorova, N.; van der Aalst, W.M.P.

*Published in:*

IEEE RCIS 2016 - IEEE 10th International Conference on Research Challenges in Information Science, 1-3 may 2016, Grenoble, France

*DOI:*

[10.1109/RCIS.2016.7549355](https://doi.org/10.1109/RCIS.2016.7549355)

Published: 23/08/2016

*Document Version*

Accepted manuscript including changes made at the peer-review stage

**Please check the document version of this publication:**

- A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

*Citation for published version (APA):*

Van Eck, M. L., Sidorova, N., & Van Der Aalst, W. M. P. (2016). Enabling Process Mining on Sensor Data from Smart Products. In IEEE RCIS 2016 - IEEE 10th International Conference on Research Challenges in Information Science, 1-3 may 2016, Grenoble, France (pp. 1-12). [7549355] Brussels: IEEE Computer Society Press. DOI: 10.1109/RCIS.2016.7549355

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

**Take down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# Enabling Process Mining on Sensor Data from Smart Products

Maikel L. van Eck\*, Natalia Sidorova, Wil M.P. van der Aalst  
Eindhoven University of Technology, The Netherlands  
Email: {m.l.v.eck,n.sidorova,w.m.p.v.d.aalst}@tue.nl

**Abstract**—In this paper we address the challenge of applying process mining to discover models of human behaviour from sensor data. This challenge is caused by a gap between sensor data and the event logs that are used as input for process mining techniques, so we provide a transformation approach to bridge this gap. As a result, besides the automatic discovery of process models, the transformed sensor data can also be used by various other process mining techniques, e.g. to identify differences between observed behaviour and expected behaviour. We discuss the transformation approach in the context of the design process of smart products and related services, using a case study performed at Philips where a smart baby bottle has been developed. This case study also demonstrates that the use of process mining can add value to the smart product design process.

**Index Terms**—process mining, smart products, product design, sensor data, activity recognition

## I. INTRODUCTION

In recent years an increasing number of smart products are being developed [1]–[5]. Such smart products are able to collect, process and produce information and they can make use of knowledge about themselves, their users and their context [1], [2]. These smart products are often combined with context-aware services to form a smart product-service system [3]. Examples of smart product-service systems include thermostats that automatically adjusts the heating of your house to your routines [4] and fitness trackers that provide exercise recommendations based on your daily activity [5].

However, developing these products and services is not an easy task [2]–[5]. It is difficult for the designers to know beforehand how a system can best be used or how it will be used in reality [2]. Yet, a thorough understanding of the users and the environment is necessary to develop smart product-service systems [3]. Therefore, it is essential to obtain this understanding during the design and development process.

An important aspect of smart products that can aid in improving user understanding is the data generated during product use, which can be collected by the product developers. This data can come from user interaction with the related services [4] or from sensors on the physical product, e.g. an optical sensor to measure heart rate or an accelerometer to detect movement on a fitness trackers [5]. Such data provides insights into the activities of people and their use of the

products [2], [4], [5], which gives designers rapid feedback on how well their products function in the real world and what elements can still be improved.

Yet, it can be a challenge to obtain the useful information from *sensor data* in order to learn how to improve the smart product design. One issue is that not all collected data is relevant, as some data may not say anything about the actual use of the product [5]. Another part of this challenge is to not only learn about a small set of observed interactions between a user and a product, but to be able to infer what this means for possible interactions in general [4]. Therefore, a necessary step of obtaining useful information is finding a way to map the collected sensor data to models of user behaviour.

Models of product user behaviour are traditionally created by hand in the product design process in order to explore possible interaction scenarios [2], [6]. However, the creation of such models takes time and they may not be an accurate reflection of what users really do when using a product. Fortunately, process mining techniques can be used to automatically discover human-readable models of behaviour based on observed behaviour [7]. There are also process mining techniques that can map behavioural data onto a pre-defined or discovered model, e.g. in order to find where product users deviate from the expected behaviour or to identify completely new behaviour [7].

There are several open research challenges that need to be tackled to enable the application of process mining in the context of sensor data, as identified by Leotta et al. [8]. Among the main challenges are: (1) mapping sensor measurements to human activities, and (2) grouping activities into process instances. These challenges need to be addressed to close the gap between sensor data and the traditional event logs assumed as input for process mining techniques, which produce output such as shown in Fig. 1.

In this paper we address the challenge of applying process mining to discover models of human behaviour from sensor data. To achieve that we provide an approach to transform sensor data into an event log that can be used as input for any process mining technique. We discuss this approach in the context of the design process of smart products and related services, using a case study performed at Philips where a smart baby bottle has been developed. Process mining generated new insights from the sensor data. We describe how these insights helped the product designers involved in the case study. Finally, we discuss some of the challenges that we encountered

\*This research was performed in the context of the IMPULS collaboration project of Eindhoven University of Technology and Philips: “Mine your own body”.

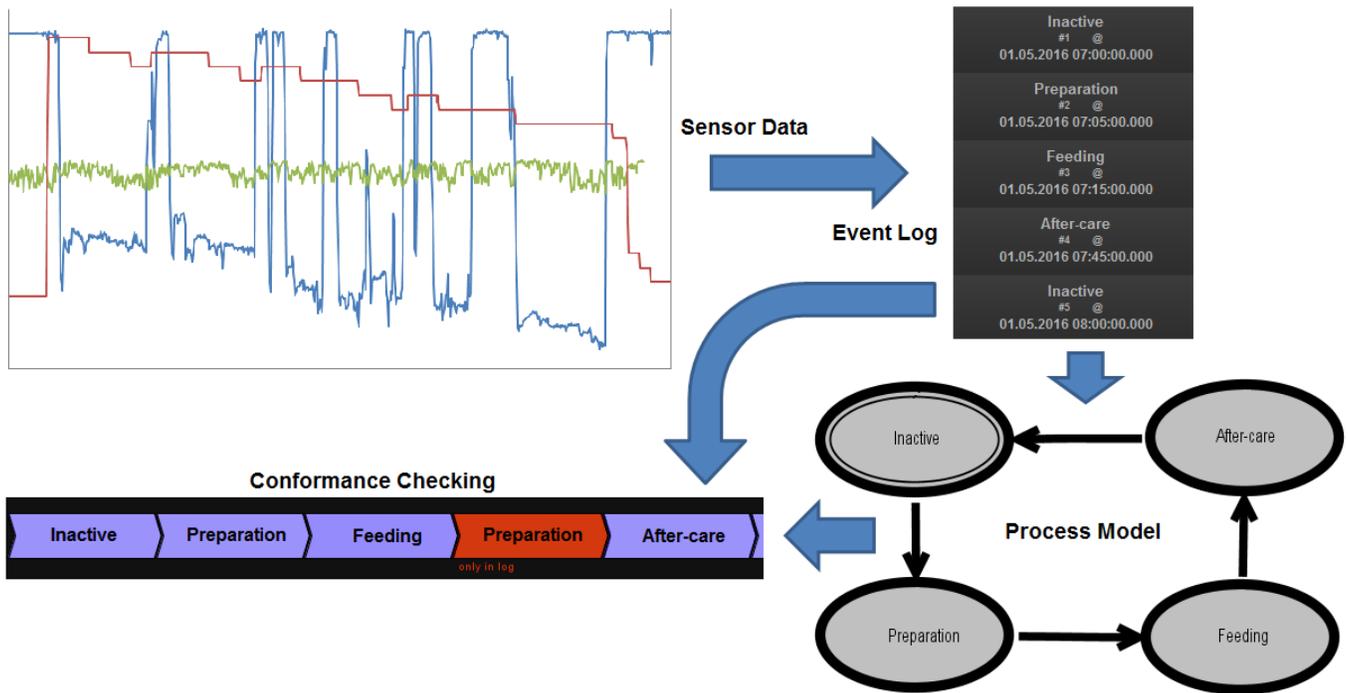


Fig. 1: An overview of the application of process mining on sensor data. Sensor data first needs to be transformed into an event log, the traditional input for process mining approaches such as process discovery or conformance checking. Process discovery techniques automatically discover process models. Conformance checking takes an event log together with a process model, in order to identify deviations between observed and expected behaviour.

when applying existing process mining techniques.

The remainder of this paper is structured as follows. In Section II related work is discussed. The approach to transform sensor data into an event log is described in Section III. The smart baby bottle design case study is discussed in Section IV. Finally, Section V concludes the paper and provides several directions of future work.

## II. RELATED WORK

In this section we present related work with respect to the two research challenges that need to be addressed to transform sensor data into event logs.

### A. Mapping Sensor Measurements to Human Activities

The challenge of mapping sensor measurements to activities can be related to two well-studied problems. The first is that of change-point detection, i.e. the detection of the points in time where the behaviour measured by the sensors changes. The second is that of activity recognition, i.e. the detection of the occurrence of activities from a sequence of observations.

The output of change-point detection approaches is a segmentation of the sensor data [9]–[11]. Within each segment the sensor data is homogeneous with similar properties, while successive segments separated by a change-point are heterogeneous. An individual segment can represent a single activity, under the assumption that the occurrence of an activity has a distinct effect on the sensor measurements. However, complex activities may cause multiple change-points in the

sensor data, resulting in a sequence of segments with specific properties. The segments identified by change-point detection techniques are also not labelled or grouped by the activities they represent [10]. Therefore, it is still necessary to create a mapping between the output of a change-point detection approach and recognisable activities.

Various approaches from the field of activity recognition produce such mappings between segments of sensor data and recognisable activities [12]–[15]. These approaches vary in complexity and in their assumptions on the input data. Most activity recognition approaches take as their input a set of time windows, segments of sensor data, or discrete events with certain features that are then classified to identify which activities were performed during these observations [12]. However, this means that these techniques often need large amounts of training data in order to learn a classifier to recognise the activities [13]. In the context of smart product design this is problematic because training data will not be available during product development due to limited numbers of prototypes and users involved in testing.

Another issue is that relatively few approaches can deal with streaming data or discover patterns indicating activities that were previously unknown, i.e. unsupervised learning [14], [15]. This last aspect is essential for activity recognition in the context of product design because it is very difficult to know beforehand exactly how people will use a product. Cook et al. present an unsupervised learning technique that is specifically

developed to discover unknown activities in discrete sensor event sequence data [15]. However, this technique is not directly applicable to continuous time series sensor data such as shown in Fig. 1, which would first need to be segmented and labelled. Another approach that has been shown to be able to discover activities in segmented sensor data is that of clustering using U-Shapelets [16]. This approach clusters segments of sensor data based on the characterising shape of the data within a segment.

Within the field of process mining there are also approaches that attempt to identify high-level activities from lower level observations. These approaches generally assume discrete events as their input. An approach by Günther et al. clusters events based on matching attributes and their proximity in time in order to create high-level activities [17]. However, this technique is limited by the requirement that the activity durations cannot be significantly longer than the time between activities. Another approach by Bose et al. aims to detect repeating sequences of events that are then grouped into a higher level activity [18]. This technique is similar to the Activity Discovery approach presented by Cook et al. [15], as it only takes into account the labels of the low-level events and the order in which they occur.

### B. Grouping Activities into Process Instances

The challenge of grouping activities into process instances is related to the problem of event correlation [7]. There are often various notions of process instances that can be chosen for a given set of activities to group them by. This choice is therefore subjective and related to the desired process perspective.

Event correlation approaches often use shared attributes to determine which activities can be correlated [19]–[21]. For example, each activity in an ordering process can be grouped by their order identifier when the desired process notion is that of the activities related to a single order, while grouping the same set of activities by the person performing the activity will result in a process notion related to the activities performed by specific people in the process. The main difference between these approaches and the challenge of grouping activities into process instances in the context of product use analysis is that event correlation is generally concerned with separating activities of overlapping process instances, while activities belonging to different instances of using a product generally do not overlap in time.

## III. TRANSFORMING SENSOR DATA INTO EVENT LOGS

In this section we discuss an approach to map sensor measurements to human activities and then group the activities into process instances. This approach combines elements from the areas of change-point detection and activity discovery as discussed in Section II. It is characterised by being able to detect potentially unknown activities in unlabelled sensor data.

We define sensor data as time series of sensor measurements, such as is shown in Fig. 1. A given data point is a

measurement belonging to a specific sensor, made at a certain moment in time, and recording the value of the measurement.

The desired output of our sensor data transformation is shown in Fig. 2). For a given collection of sensor data, the goal is to identify the activities that are performed during a specific period of time. Each activity has a start and end time, and a label. Activities may overlap, e.g. ‘Use bottle’ overlaps with the other activities, and they can be related to data from a single sensor, e.g. ‘Add Food’ and the temperature sensor, or related to multiple sensors, e.g. ‘Use bottle’.

The approach consists of the following steps that we discuss in more detail in the remainder of this section:

- Segmentation of sensor data
- Feature selection and calculation to characterise segments
- Clustering of segments to identify time periods with similar behaviour
- Interpretation of clustering results to label behaviour
- Activity creation by grouping labelled segments
- Process instance creation

### A. Sensor Data Segmentation

To transform the sensor data into activities, the sensor measurement time series are segmented into small windows. These small windows are later labelled such that ‘similar’ windows receive the same label. This is shown in Fig. 3. The labelled windows provide information on change-points per sensor, from which activities can be inferred. This window-based segmentation is a basic bottom-up change-point detection technique that identifies a change-point whenever two consecutive windows receive different labels [9]. However, more advanced segmentation techniques can also be used with our approach.

The segmentation of the sensor data is straightforward. A specific window size is chosen, expressed as a period of time, e.g. 1 second. The time series of sensor data is then divided into windows with a duration equal to the chosen window size, so that every sensor measurement is assigned to a specific window.

If the sensor data contains measurements from multiple sensors then they can either be segmented jointly as a single time series or the time series of each individual sensor can be segmented separately. In most cases it is better to segment the sensor data for different sensors separately. The main reason for this is that this segmentation is used to detect where the sensor data changes and multivariate change-point detection is non-trivial [11]. The later steps of the transformation approach aim to characterise the sensor data within each segment in order to map it to an activity and this characterisation is much more complex for multivariate data. The exception to this is when sensors are known to be highly correlated, e.g. different directional components of an accelerometer in a symmetric product, in which case it is advised to segment the combined measurements.

When segmenting sensor data with measurements from multiple sensors, a window contains all measurements made during that time window, irrespective of the sensor from which

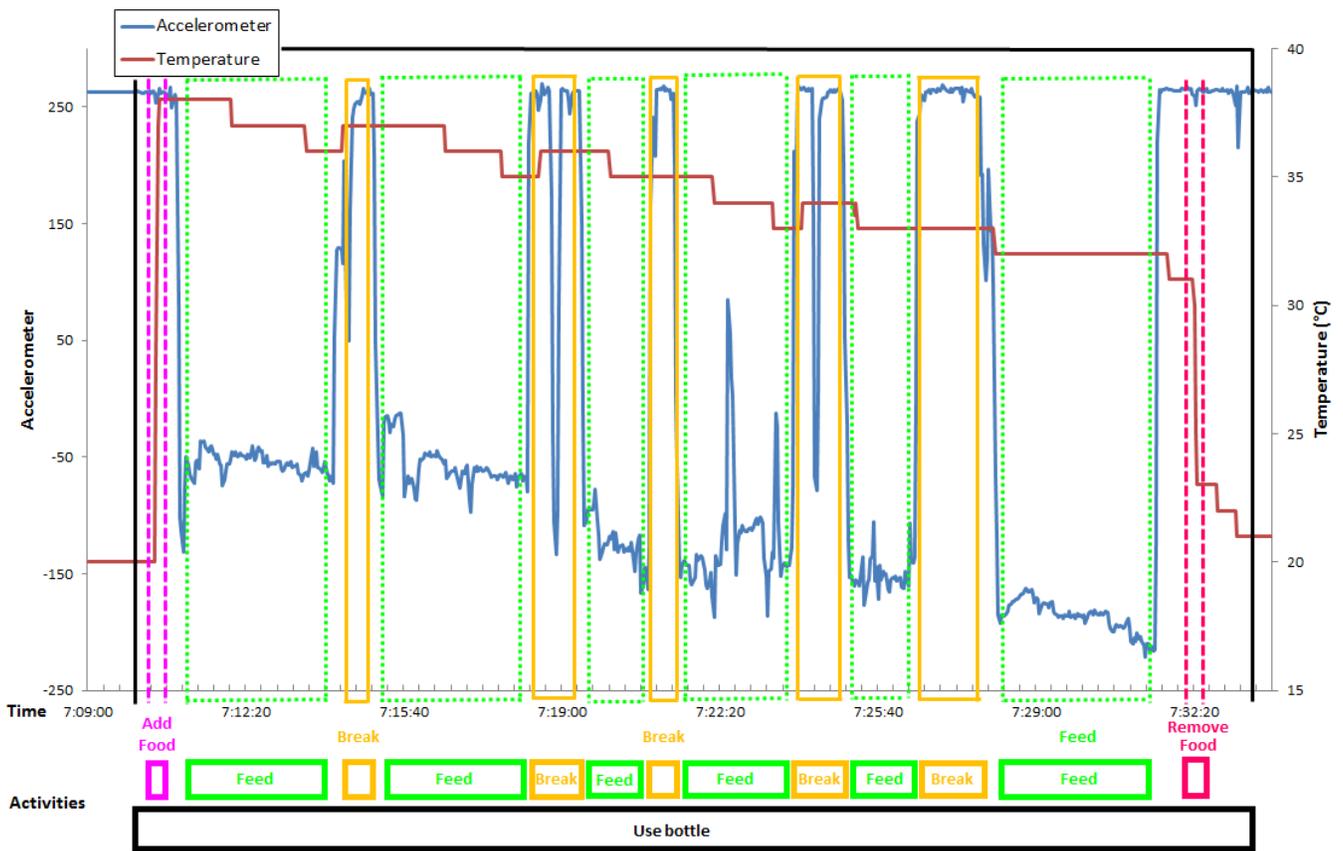


Fig. 2: A set of activities identified in the accelerometer and temperature sensor data from Fig. 1. Each activity is characterised by a specific pattern in the data of one or more sensors, e.g. *Add food* is characterised by an increase in temperature.

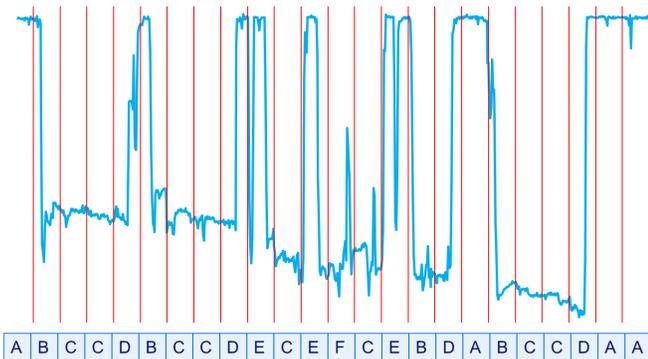


Fig. 3: Segmented and labelled sensor data. Each segment is labelled based on the properties of the sensor measurements it contains, e.g. a stable high sensor signal is labelled with ‘A’. Change-points in the sensor data result in adjacent segments having different labels.

they originated. When the sensor data is segmented separately for each sensor, each window only contains measurements from a single sensor. In that case the time series of different sensors may even be segmented with different window sizes, e.g. depending on the sampling rate of each sensor.

The only parameter of the segmentation itself is the window size. The optimal size of the individual windows depends mainly on the sampling rate of the sensor and the expected duration of activities [12]. There is a trade-off between having enough measurements to accurately detect the type of behaviour in a window and the possibility of having multiple changes in behaviour within a window. The challenge here is that if the sampling rate of the sensors is low then a large window size reduces the effects of noise in the measurements. On the other hand, increasing the window size increases the chance that a single window contains multiple activities and that makes the activity detection more difficult.

### B. Segment Labelling

After segmentation, each window of sensor measurements needs to be labelled. This is done by calculating relevant features for each segment based on its data measurements and subsequently clustering the segments. Each cluster is then labelled based on their characterisation using domain knowledge.

There are various of features that can be calculated based on the data points in a given segment [9], [14]. These include simple properties of the sensor measurements such as the average sensor value within the segment or the rate of change,

TABLE I: A characterisation and labelling of cluster centroids of accelerometer data based on six features. The sensor data has three dimensions (x, y and z) and measurements range from -300 to 300 for each dimension.

Cluster Label	Median X	$\Delta X$	Median Y	$\Delta Y$	Median Z	$\Delta Z$	Domain Expert Interpretation
A	-11	0,1	8,3	0,0	252	5,3	Upright Unmoving
B	-146	-161	-94	-93	-95	-199	Downturned Bigmove
C	-56	-4,6	-114	-4,4	-123	-4,2	Downturned Unmoving
D	-24	-172	11	43	230	329	Upright Bigmove
E	-10	96	26	125	228	327	Upright Bigmove
F	170	164	-31	-62	-90	-183	Downturned Bigmove
G	-7,0	25	-1,4	-175	231	321	Upright Bigmove
H	30	3,8	169	20	-110	16	Downturned Unmoving

or domain specific features. Frequency domain features such as spectral power are also frequently used in activity detection approaches. Determining which features to calculate depends mainly on the relation between the sensor measurements and the activities that are expected.

After calculating the features for each segment, a clustering approach is used to generate implicit labels for each sensor data segment, as shown in Fig. 3. For clustering approaches parameterised with the number of desired clusters, this parameter should be higher than the expected number of activities. Different clustering algorithms can be used, but centroid-based clustering techniques such as k-means [22] make it easier to interpret the meaning of the clusters and their relation to activities.

Based on the interpretation of the properties of each cluster, a domain expert is required to assign a meaning to the clusters. For centroid-based clustering techniques this can be done by looking at the features of the cluster centroids and determining what activities are expected to be represented by such features.

An example for the accelerometer data from Fig. 3 is shown in Table I. The characterising features of the centroid of each cluster label are shown. This shows that segments assigned a label characterised by having a positive ‘Median Z’ are interpreted as time periods where the product was upright, while a negative value indicates it was downturned. A large ‘ $\Delta$ ’ value for a dimension indicates a big move of the product in that direction, so labels characterised by having low ‘ $\Delta$ ’ values for all three dimensions are interpreted as representing the time when the product is unmoving.

### C. Multi-sensor Activity Creation

After the interpretation of the sensor segment clustering the segments can be grouped together to create activities. For a single segmentation this is straightforward, as adjacent segments with the same label can be merged into larger segments. It is also possible to add noise filtering in this step, e.g. if multiple segments with the same label are separated by a single segment with a different label then it can be filtered out.

In the case where multiple sensor signals have been segmented separately the segments need to be synchronised before they can be grouped into activities, especially when they were segmented with different window sizes. The approach to synchronise two segmented and labelled sequences with different window sizes is shown in Fig. 4. A new segmentation

is created with a window size equal to the smallest window size of the segments. The labelling of this new segmentation is done on the basis of the largest overlap between the new segments and the old segments. For example, the leftmost new segment labelled *C2* in Fig. 4 fully overlaps with segment *C* and more with segment 2 than 1.

The result is a collection of activities for a group of sensors. However, depending on the interpretation of the clustering results they may represent patterns in the sensor data rather than true human activities that resulted in these patterns. This can be addressed by another round of relabeling using domain knowledge, e.g. segment labelled as having a high temperature and a downturned accelerometer indicate the start of a ‘Feed’, as shown in Fig. 2. Alternatively, some activity detection techniques can be used to create higher level activities [15], but those will also need to be interpreted and labelled.

### D. Process Instance Creation

As discussed in Section II-B, there exist several techniques to group activities into process instances. Most of these techniques rely on additional attributes of activities to relate them into distinct process instances, e.g. an order identifier to link all activities belonging to the same order. However, such additional attributes are generally not available for the activities we consider here because they were created from sensor data.

Therefore, it is important to consider the process that we are interested in and to determine what characterises an instance of that process. When the process of interest is the interaction of a product user with a smart product then each interaction forms an instance of that process. This means that activities can be divided into process instances if the start or end of such an interaction can be identified, e.g. due to the presence of a specific activity or the product returning to a state of inactivity. Other possibilities include interactions having a maximum duration or happening at a specific time of the day. Hence, the correct choice for determining what characterises a process instance depends mainly on domain knowledge.

## IV. CASE STUDY

The approach to enable process mining on sensor data was used in a case study where Philips worked on the design of a smart baby bottle equipped with various sensors. The goal of the study was to investigate the characteristics of the data that would be generated during the use of the bottle, and to explore

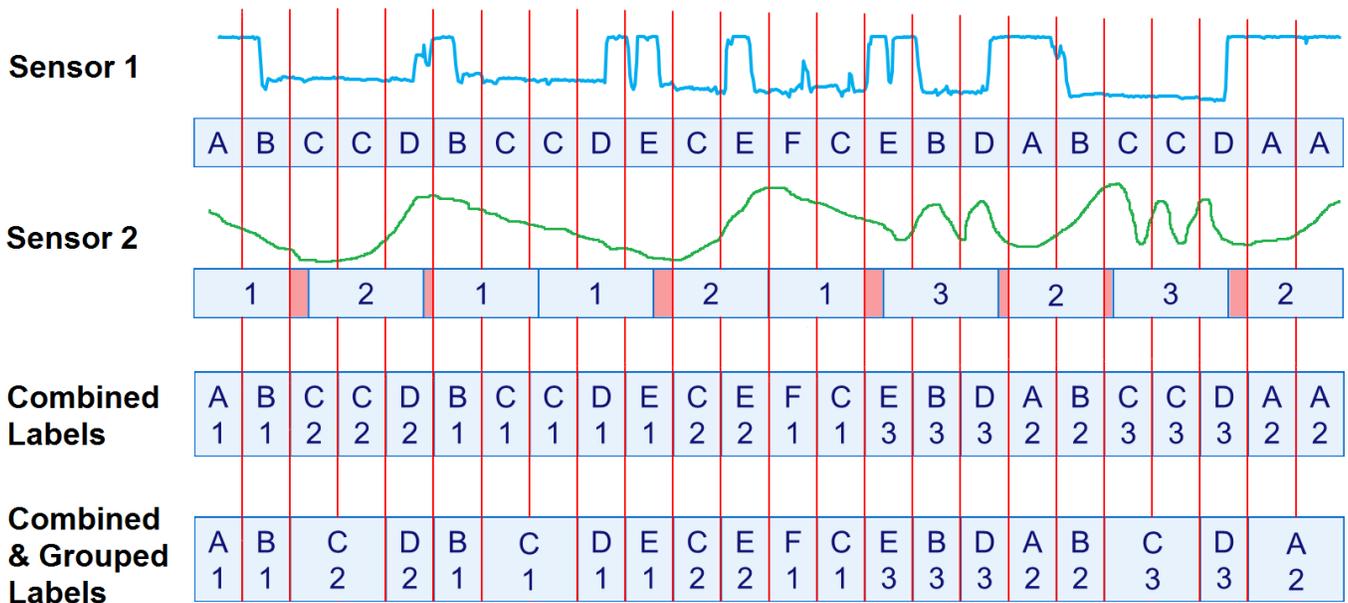


Fig. 4: The synchronisation of two sequences of segments created with different window sizes. The segments from sensor 2 have a larger window size and are not properly aligned with the windows from sensor 1.

what product improvements could be made and what related services could be offered based on analysis of this data.

As part of the design process, the product designers manually created a model of the intended product use, shown in Fig. 5. This model consists of three main parts, reflecting the preparation of the feeding, the actual feeding and the aftercare.

#### A. Sensor Data Transformation and Process Instance Creation

There were 9 families participating in the study, who were each provided with a prototype of the smart bottle for a 3 week period. 358 instances of baby feedings are contained in the sensor data gathered during the study. Each feeding instance was confirmed by the participants through an app where post-feeding feedback could be filled in. The designers also created a small dataset with a known mapping between the sensor data and user behaviour by performing activities with a prototype in a lab setting and annotating the data manually.

The smart bottles were equipped with four different types of sensors. Specifically, a temperature sensor, a 3D accelerometer, a light intensity sensor, and a sound level sensor. The sampling rate of these sensors was around 7 Hz, so the sensor data was of low granularity.

After collecting the sensor data, an event log was created as discussed in Section III. This event log was used as input for various process mining techniques in order to obtain insights for the product designers.

1) *Sensor Data Segmentation*: The sensor data from each of the four sensor types was segmented and labelled separately. This was done because interpreting the clustering results of segments of multi-sensor data proved to be too difficult. In addition, classifiers learned using the multi-sensor clustering results for the small labelled dataset had a very low accuracy.

The accelerometer data was processed slightly differently than data from the other sensors because it consisted of three directional components ( $x$ ,  $y$  and  $z$ -dimensions). The directional components form three distinct time series of sensor measurements that were segmented and labelled together, meaning that features calculated for a window of accelerometer data may be based on any or all of the component measurements. The reason for combining all directional components was that they are highly correlated for certain types of movement of the bottle.

A window size of 2 seconds was chosen for the segmentation, based on the detection of changing behaviour in the small labelled dataset that was generated by the designers. A smaller window size, e.g. 1 second, resulted in a large number of windows containing only one or two sensor measurements, due to non-uniform data recording, and hence unreliable features. On the other hand, larger window sizes resulted in an increasing number of windows containing multiple different product user actions. The chosen segmentation window size represents a good trade-off between these aspects for the labelled dataset.

2) *Segment Labelling*: The features that were calculated for each window were based on the detection of behaviour in similar datasets [9], [14] and domain knowledge for this study. These features consisted of the median of the sensor values and rate of change within the window, the relative change in sensor values compared to a time period before the window, and energy and power spectrum features from the frequency domain of the sensor signal. The windows of the accelerometer data had these features calculated for each of the three dimensions.

The clustering of the sensor data windows was done using k-means clustering. The windows for each specific sensor type

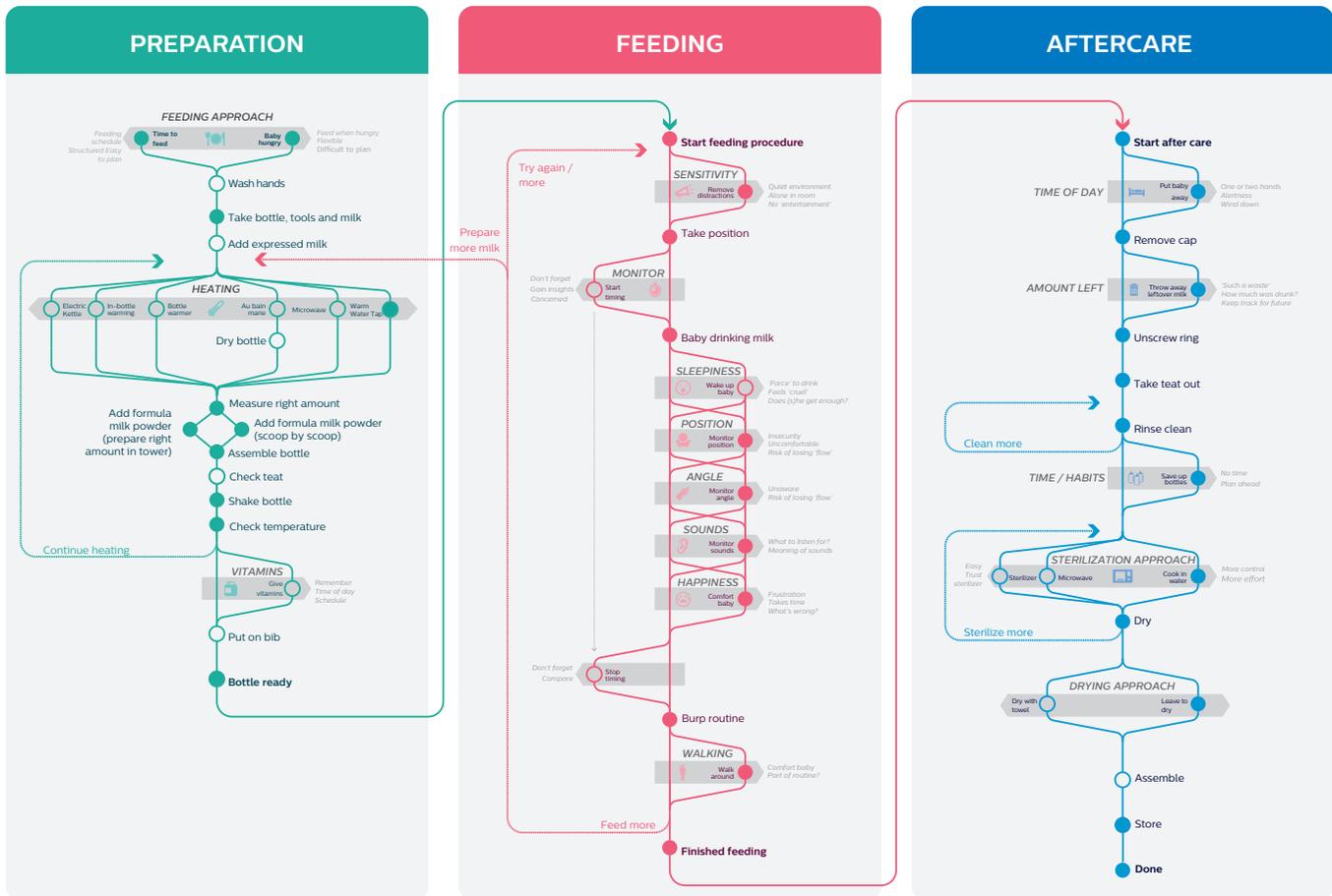


Fig. 5: A model of product user behaviour for the smart baby bottle, manually created by product designers. The feeding process consists of three main stages: preparation, feeding and aftercare, denoted in green, red and blue, respectively.

were clustered into 16 clusters. The number of clusters was determined iteratively and based on interpretation of the cluster centroids using domain knowledge. The use of U-Shapelet clustering was also investigated, as it has been used in other studies with similar data to discover different types of activities from sensor data [16]. However, evaluation on the labelled dataset showed that this technique does not separate different activities into different clusters. This may be related to the sensor sampling rate in our case study, which is orders of magnitude lower than in comparable studies.

The properties of the cluster centroids that were obtained with k-means clustering were subsequently studied in order to determine labels for the clusters based on domain knowledge. This resulted in 7 different labels for the temperature data, 4 labels for the accelerometer data, and 6 labels for the light and sound data. Examples of such labels are *Upright\_Unmoving*, for accelerometer data indicating that the bottle is not being used, and *Warm\_BigIncrease*, for a spike in the temperature suggesting that warm food is being put into the bottle.

3) *Multi-sensor Activity Creation:* For this case study the synchronisation of the data from different sensor types was trivial. This is due to the simultaneous recording of measurements for all sensors at every measurement point

and the segmentation with uniform window sizes. Therefore, the windows of different sensor types were already aligned. Classifiers learned using the resulting multi-labelled windows had a classification accuracy of around 85% for the labelled dataset.

Adjacent windows with the same label were grouped into bigger intervals and no additional noise filtering was applied. Windows were only considered to have the same label if their labels for all four sensor types agreed.

Note that the resulting activities have labels that generally describe the state of the different sensors for the duration of the activity, instead of a label recognisable as a human activity. Applying process mining techniques on this event log of low-level sensor states gave more insights than first relabelling and aggregating the different states to more general descriptions of activities.

4) *Process Instance Creation:* The process instance creation for this case study was also straightforward. As shown in Fig. 5, the designers considered in their user behaviour model the process of a single feeding and the activities that occur before and after the feeding. Therefore, we created an event log where each process instance contains one feeding instance, starting with the bottle being inactive and ending with

the bottle returning to the inactive state. The inactive state is easily detected and the feeding instances are characterised by an increased temperature and specific types of motion.

### B. Discovering Models of Smart Bottle User Behaviour

One of the most well-known applications of process mining is the automatic creation of fact-based process models, or process discovery [7]. There are numerous process discovery techniques and they produce different types of models. One type of model that can be mined is the end-to-end process model, which highlights the main paths that can be taken when executing a process instance from start to end. The event log created from the smart bottle sensor data was mined for end-to-end process models and one result is shown in Fig. 6. Other types of process models only show specific parts of a process, indicating potentially interesting scenarios that have been observed when the process was executed. Two examples of such discovered scenarios are shown in Fig. 7.

1) *End-to-end User Interaction Model*: The model in Fig. 6 shows the activities from the event log and the order in which they occur during a feeding instance. Note that the model only shows a generalisation of the behaviour that was observed, i.e. only the activity flows that were observed frequently are shown as possible paths through the model. At a higher level of abstraction, different areas of the discovered model can be mapped to the conceptual model of the designers from Fig. 5. The area of the model highlighted in green shows the bottle being cold and generally inactive. After that the orange area, enlarged in Fig. 8, shows the next step of preparation phase, which is the heating of the bottle. The feeding process itself is shown in red and in more detail in Fig. 9.

The model from Fig. 6 was obtained using only the data related to the accelerometer and temperature sensors. Including the light and sound sensor data did not aid in the discovery of an understandable model for the feeding process. In general, there were very few relations between light and sound data and different types of behaviour.

One important difference from the designers model is that the aftercare phase, shown in blue in Fig. 5, is not present in this mined model. This is partially related to the mining algorithm used to discover the model, the Inductive visual Miner [23]. This algorithm, like many others, has difficulties incorporating duplicate activities in the model and many forms of motion occurring during the preparation phase are similar to those occurring during the aftercare.

However, the most important difference between the models is that they are not made at the same level of abstraction. The model created by the designers describes activities that are clearly recognisable and understandable, even for a person without understanding of the specifics of the product. The discovered models on the other hand describe activities based on their measurable effects on the sensor data. This is the result of the way in which the event logs that were used as input were constructed.

Although the mined model from Fig. 6 has several issues and differs from the designers model shown in Fig. 5, it

still provides valuable insights. These insights are related to both the limitations of the collected data and to potential improvements of the product.

As mentioned above, the aftercare phase is not present in the discovered model, but this is not only related to the mining algorithm that was used. Many of the activities from the aftercare phase are simply not recorded in the current sensor dataset. For example, the cleaning, sterilising and drying of the bottle is generally not visible because the participants removed the part of the smart bottle containing the sensors before engaging in these activities. Similarly, the shaking of the bottle during the preparation phase was not observed in the data, although it was during an earlier, smaller, test study with the same product.

This shows that it is important for the designers of a smart product to be aware of and study the activities that can or cannot be detected from the data. When certain activities are considered to be important but cannot be detected, then the first step is to find out why this is the case. It may be that the product users are not using the product as intended, e.g. shaking the bottle without the sensor component attached, in which case the designers will need to come up with a way to steer the user towards intended behaviour. If technical limitations are the cause, e.g. the sensors being unable to detect sterilisation in the microwave, then other sources of data, e.g. the microwave itself, or extensions to the product may be necessary.

Another interesting observation that can be made from the mined model is highlighted in green in Fig. 9. The model shows that there is a correlation between moving the bottle upright during the feeding and a small temperature increase. The explanation for this correlation, which was initially surprising for the developers, could be found using their domain knowledge. Namely, the temperature sensor was located at the bottom of the bottle and during the feeding the food is no longer in contact with the bottom of the bottle. Hence, the temperature sensor does not provide an accurate reading of the temperature of the food during the feeding, which is important for statistics or recommendations on reheating the food. Possible solutions for this problem include placing an additional sensor at the tip of the bottle or only considering the sensor to be measuring the food temperature when the bottle is upright. This example again shows that process mining can provide valuable insights on the design of a smart product and related services.

The model in Fig. 9 also shows which parts of the model can be improved in quality. The red parts highlight the places where feeding instances occasionally deviate from the behaviour described by the model. Many of these deviations are related to the lack of an aftercare phase in the mined model. The instances where the data did show the presence of aftercare activities, e.g. a temperature drop due to the bottle being emptied, therefore deviate from the model. A smaller number of deviations were related to the cooling of the bottle during long feeds and subsequent reheating of the food, such as shown in the instance in Fig. 11. The underlying cause for

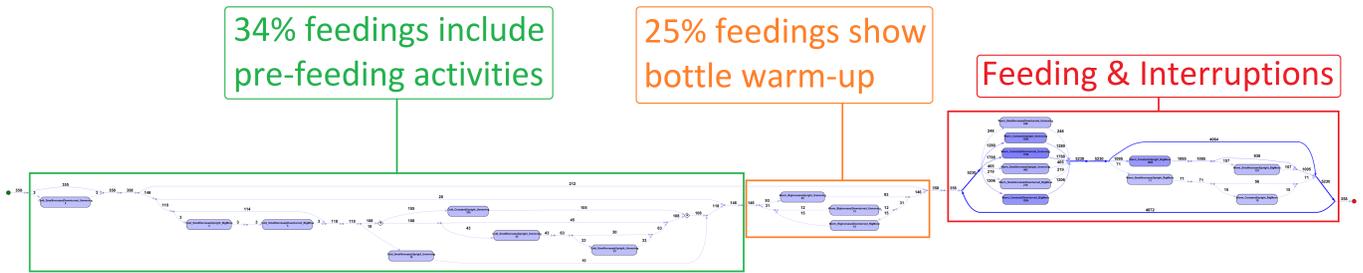


Fig. 6: A discovered end-to-end process model for the use of the smart bottle, showing which activities are performed and in what order. The area highlighted in green relates to the bottle being cold and mostly stationary. The orange part relates to the bottle being heated and is shown in Fig. 8. The red part is shown in Fig. 9 and relates to the feeding of the baby.

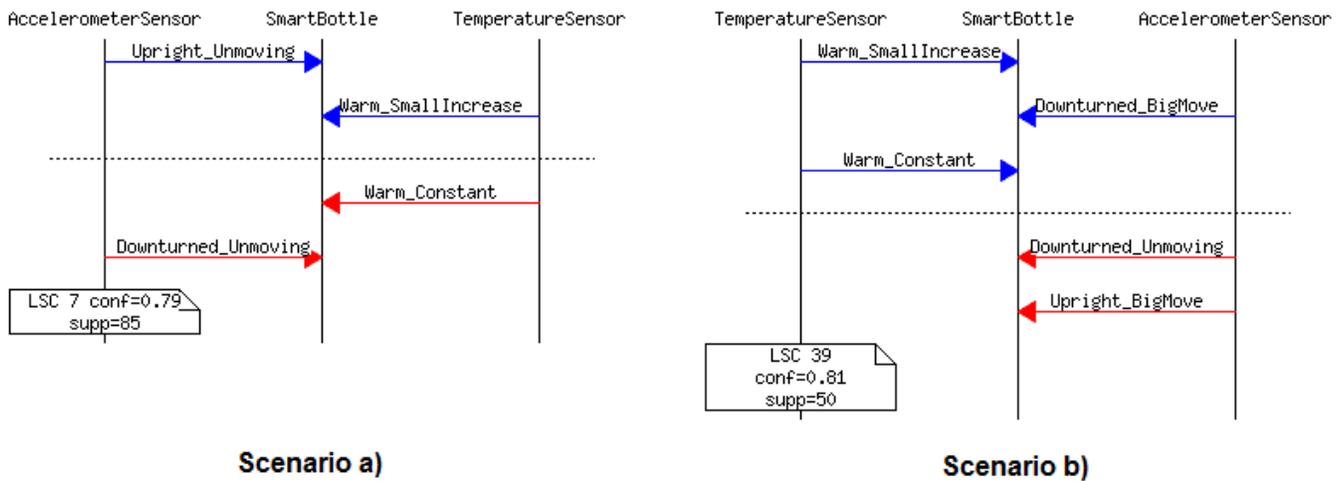


Fig. 7: Two discovered scenarios denoting that if the blue activities are observed then the red activities are expected to occur as well. Scenario a) shows that heating the bottle results in a feeding. Scenario b) describes that the start of a feeding results in a feeding that eventually ends.

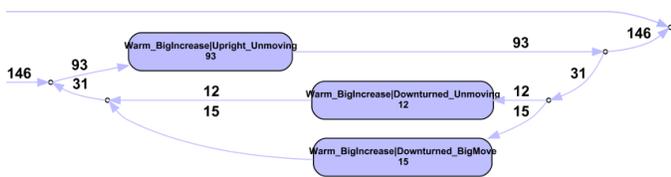


Fig. 8: The activity *Warm\_BigIncrease|Upright\_Unmoving* indicates that the bottle has been heated. This is sometimes seen to be followed by shaking the bottle (*Warm\_BigIncrease|Downturned\_BigMove*) or checking the temperature (*Warm\_BigIncrease|Downturned\_Unmoving*). These steps are described to be part of the preparation phase shown in Fig. 5.

these deviations is related to activities that occur in distinct parts of the process and that can have a different interpretation based on their context, e.g. heating and reheating of the bottle. These deviations show that mining good models can still be challenging, especially when context information is necessary

to understand the real meaning of activities.

2) *User Interaction Patterns*: In addition to mining end-to-end process models such as the one shown in Fig. 6, process mining techniques were also used to discover specific scenarios or patterns [24]. Two examples of such scenarios are shown in Fig. 7.

For this case study the discovered scenarios only revealed patterns that could also be inferred from the end-to-end process model. However, interpretation of small scenarios can be easier than the interpretation of a large process model. This is especially true in this case, due to the artificial labelling of the clusters of different sensor data.

As shown in Fig. 7 and Fig. 8, the labels for behaviour detected in the sensor data are still low-level. In some cases there is a simple mapping between the artificial labels and recognisable activities, such as *Warm\_BigIncrease|Upright\_Unmoving* indicating that the food in the bottle has been heated. Unfortunately, this mapping is often more complex and an activity performed by a product user results in multiple changes in the sensor data. For example, the start of the feeding is generally marked by the occurrence of a *Downturned\_BigMove* event

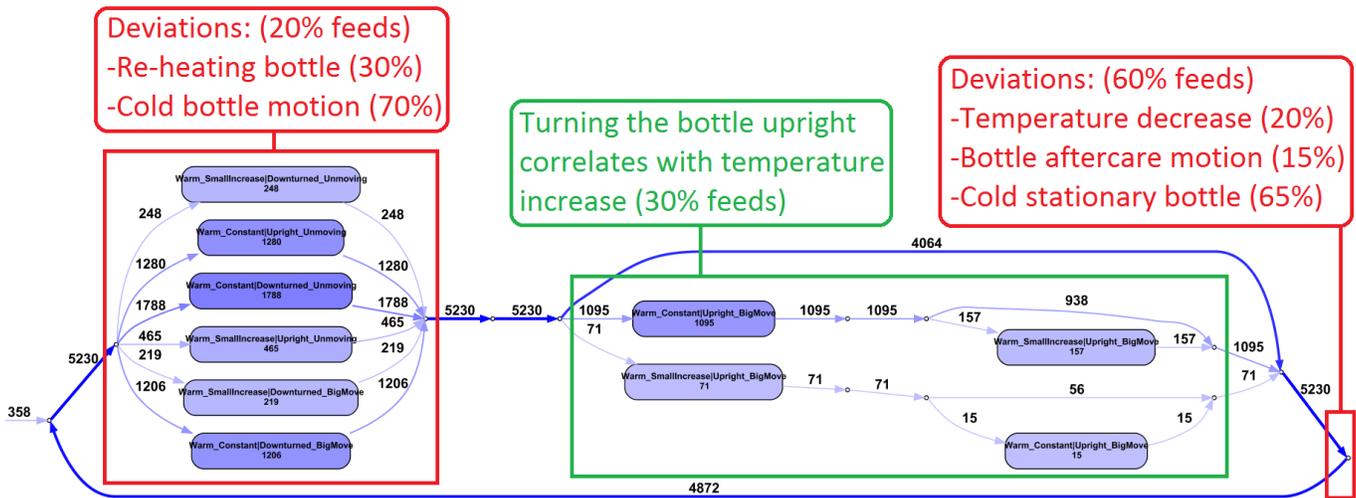


Fig. 9: A discovered model of the feeding process. The activities on the left are related to feeding and interruptions. The deviations indicate incomplete areas of the model where additional behaviour was observed in the data.

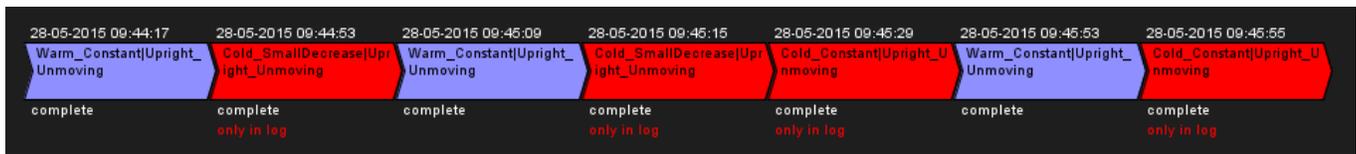


Fig. 10: A sequence of non-feeding data, recorded in between feedings, that is shown to deviate from the feeding model. There was no movement of the bottle, but small temperature fluctuations resulted in alternating labels for the sensor data.

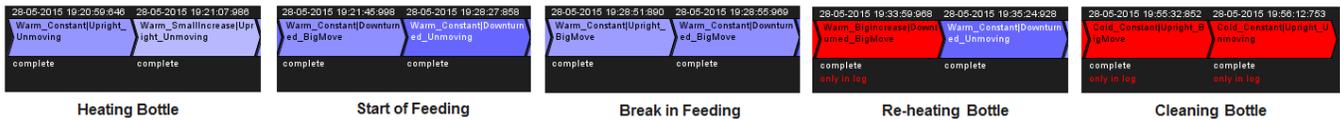


Fig. 11: A feeding instance that partially conforms to the feeding model. It contains activities indicating the heating of the bottle and the start of the feeding, as well as some interruptions in the feeding. However, the instance also contains behaviour that is not captured by the model, i.e. a re-heating of the bottle in the middle of the feeding instance and cleaning of the bottle at the end of the feeding.

followed by a *Downturned\_Unmoving* for the accelerometer while at a *Warm\_Constant* state for the temperature. These scenarios can therefore help to investigate how different user activities can be composed of multiple sensor events and also aid in the discovery of new types of behaviour that the designers have not anticipated beforehand.

### C. Detecting Noisy Behaviour

Conformance checking, another area of process mining, can be used to see the differences between process instances and a given model [7]. For each process instance, these techniques identify exactly which parts of the observed behaviour fit the model, which parts of the observed behaviour cannot be explained by the model, and which parts of the modelled behaviour has not been observed. This can be used to provide insights on the individual instances and whether they are really cases of product use or just observations of noise.

For example, although the model shown in Fig. 6 contains activities related to the preparation phase, such as the heating of the food, only 25% of the feeding instances actually contain this behaviour. This is again related to the visibility of activities in the sensor data, but conformance checking can show exactly which users or feeding instances deviate from the expected behaviour. The designers can then follow up and contact these specific users to find out why their behaviour differs from the expected behaviour. For example, in this case the participants only attached the sensor component to the bottle after heating up the food and putting it in the bottle, which is why the majority of the feeding instances go directly from an inactive to a warm bottle.

Conformance checking can also be used to identify data that does not match the expected behaviour at all, i.e. noisy behaviour that is not caused by real product use. The data gathered during the study contains 358 instances of feeding

confirmed by the participants, but data was also collected during e.g. transportation of the bottle, such as the sequence of activities shown in Fig. 10. This non-feeding data has been processed in the same way as the feeding data and was then mapped to the mined model from Fig. 6 using conformance checking techniques. The non-feeding data had a total fitness score of 22.7% and only 20 out of 813 cases (2.5%) perfectly matched the model. By contrast, the confirmed feeding data had a total fitness score of 82.3% and 101 out of 358 cases (28%) perfectly matched the model. Furthermore, several of the process instances from the non-feeding data that did fit the model on inspection appeared to be real feeding instances that were simply not confirmed by the product user as a feeding in the app. This result shows that process mining can be used to identify behaviour that is either noise or significantly different from the expected product use.

## V. CONCLUSION

In this paper we have described an approach that addresses two of the main challenges of applying process mining techniques in the context of sensor data: (1) mapping sensor measurements to human activities, and (2) grouping activities into process instances. This approach can be used to transform sensor data into an event log that can be used as input for any process mining technique.

The approach has been used in a case study in collaboration with Philips. During this case study Philips worked on the design of a smart baby bottle equipped with various sensors. The sensor data collected in the case study were transformed into event logs and subsequently analysed using a number of different process mining techniques.

This process mining analysis provided a number of insights for the designers of the smart product and its services. For example, information on the effectiveness of the placement of sensors and the ability to infer certain types of desired or unexpected user behaviour from the data. This showed not only the value of the approach in the context of smart product design, but also the value of process mining in general in this context.

Future work includes additional research on both the extension of the transformation approach and its application. For example, the interpretation of the activities created by the approach currently requires significant domain knowledge regarding the link between sensor measurements and possible user behaviour. Also, the technique was applied during the design of a single isolated smart product, but with more and more smart products being developed their interactions in a bigger connected environment will also need to be discovered and taken into account during the design process.

## ACKNOWLEDGEMENTS.

We would like to thank Berkay Buharali, Peter Lövei, Janne van Kollenburg, Sander Bogers and all other Philips employees involved for their collaborative efforts.

## REFERENCES

- [1] S. A. Rijdsdijk and E. J. Hultink, "How today's consumers perceive tomorrow's smart products\*," *Journal of Product Innovation Management*, vol. 26, no. 1, pp. 24–42, 2009.
- [2] E. Aitenbichler, F. Lyardet, G. Austaller, J. Kangasharju, and M. Mühlhäuser, "Engineering intuitive and self-explanatory smart products," in *Proceedings of the 2007 ACM Symposium on Applied Computing (SAC), Seoul, Korea, March 11-15, 2007*, 2007, pp. 1632–1637.
- [3] A. Valencia, R. Mugge, J. P. L. Schoormans, and H. N. J. Schifferstein, "The design of smart product-service systems (psss): An exploration of design characteristics," *International Journal of Design*, vol. 9, no. 1, 2015.
- [4] R. Yang and M. W. Newman, "Learning from a learning thermostat: lessons for intelligent systems for the home," in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM, 2013, pp. 93–102.
- [5] B. Lowens, V. Motti, and K. Caine, "Design recommendations to improve the user interaction with wrist worn devices," in *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. IEEE, 2015, pp. 562–567.
- [6] L. L. Constantine, "Human activity modeling: Toward A pragmatic integration of activity theory and usage-centered design," in *Human-Centered Software Engineering - Software Engineering Models, Patterns and Architectures for HCI*, 2009, pp. 27–51.
- [7] W. M. P. van der Aalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer, 2011.
- [8] F. Leotta, M. Mecella, and J. Mendling, "Applying process mining to smart spaces: Perspectives and research challenges," in *Advanced Information Systems Engineering Workshops*. Springer, 2015, pp. 298–304.
- [9] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, and P. J. M. Havinga, "Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey," in *ARCS '10 - 23th International Conference on Architecture of Computing Systems 2010, Workshop Proceedings, February 22-23, 2010, Hannover, Germany*, 2010, pp. 167–176.
- [10] M. H. Nguyen, Z. Lan, and F. D. la Torre, "Joint segmentation and classification of human actions in video," in *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, 2011, pp. 3265–3272.
- [11] M. Yamada, A. Kimura, F. Naya, and H. Sawada, "Change-point detection with feature selection in high-dimensional time-series data," in *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, 2013.
- [12] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, and I. Rojas, "Window size impact in human activity recognition," *Sensors*, vol. 14, no. 4, pp. 6474–6499, 2014.
- [13] E. M. Tapia, S. S. Intille, and K. Larson, "Activity recognition in the home using simple and ubiquitous sensors," in *Pervasive Computing, Second International Conference, Pervasives 2004, Vienna, Austria, April 21-23, 2004, Proceedings*, 2004, pp. 158–175.
- [14] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 3, pp. 1192–1209, 2013.
- [15] D. J. Cook, N. C. Krishnan, and P. Rashidi, "Activity discovery and activity recognition: A new partnership," *IEEE T. Cybernetics*, vol. 43, no. 3, pp. 820–828, 2013.
- [16] L. Ulanova, N. Begum, and E. J. Keogh, "Scalable clustering of time series with u-shapelets," in *Proceedings of the 2015 SIAM International Conference on Data Mining, Vancouver, BC, Canada, April 30 - May 2, 2015*, 2015, pp. 900–908.
- [17] C. W. Günther and W. M. P. van der Aalst, "Mining activity clusters from low-level event logs," in *BETA Working Paper series, vol. WP 165*, 2006.
- [18] R. P. J. C. Bose and W. M. P. van der Aalst, "Abstractions in process mining: A taxonomy of patterns," in *Business Process Management, 7th International Conference, BPM 2009, Ulm, Germany, September 8-10, 2009. Proceedings*, 2009, pp. 159–175.
- [19] H. R. M. Nezhad, R. Saint-Paul, F. Casati, and B. Benatallah, "Event correlation for process discovery from web service interaction logs," *VLDB J.*, vol. 20, no. 3, pp. 417–444, 2011.

- [20] S. Rozsnyai, A. Slominski, and G. T. Lakshmanan, "Discovering event correlation rules for semi-structured business processes," in *Proceedings of the Fifth ACM International Conference on Distributed Event-Based Systems, DEBS 2011, New York, NY, USA, July 11-15, 2011*, 2011, pp. 75–86.
- [21] R. Engel, W. M. P. van der Aalst, M. Zapletal, C. Pichler, and H. Werthner, "Mining inter-organizational business process models from EDI messages: A case study from the automotive sector," in *Advanced Information Systems Engineering - 24th International Conference, CAiSE 2012, Gdansk, Poland, June 25-29, 2012. Proceedings*, 2012, pp. 222–237.
- [22] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.
- [23] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, "Exploring processes and deviations," in *Business Process Management Workshops - BPM 2014 International Workshops, Eindhoven, The Netherlands, September 7-8, 2014, Revised Papers*, 2014, pp. 304–316.
- [24] D. Fahland, D. Lo, and S. Maoz, "Mining branching-time scenarios," in *2013 28th IEEE/ACM International Conference on Automated Software Engineering, ASE 2013, Silicon Valley, CA, USA, November 11-15, 2013*, 2013, pp. 443–453.