A-Posteriori Detection of Sensor Infrastructure Errors in Correlated Sensor Data and Business Workflows

Andreas Wombacher
Database Group,
University of Twente,
Enschede, The Netherlands
Email: a.wombacher@utwente.nl

***Abstract*—Sensor data can be interpreted as a view on physical objects effected by business processes. Since both sensor infrastructures and business workflows must deal with imprecise information, the correlation of sensor data and business workflow data might be used a-posteriori to determine the source of the imprecision. In this paper, information theory based approach is presented to distinguish sensor infrastructure errors from inhomogeneous business workflows. This approach can be applied on detecting imprecisions in the sensor infrastructure, like e.g. sensor errors or changes of the sensor infrastructure deployment.**

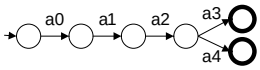
I. Introduction

More and more sensor data within an enterprise and outside an enterprise is available. Examples are fleet management and package tracking related to logistics processes, where tracking information of shipments by truck or by ship outside an enterprise is provided, while bar code and RFID readers are facilitated within an enterprise [1].

Sensor data as well as enterprise information systems describe a view on the physical world consisting of physical objects with properties and a location. The sensor data view is potentially imprecise [2]. For example, a physical object may pass by an RFID reader too fast, such that the reading can not be completed. Thus, the moving object is not observed. An enterprise information system which controls and documents the execution of business workflows - further called workflow system - is potentially imprecise [3]. For example, the workflow in the physical world may deviate from the workflow specified in a workflow system. Discrepancies can be either exceptional, i.e., ad hoc changes in the real world on request which are not reflected in the workflow system, or structural, i.e., the deviation/evolution of the implemented workflow and the workflow in the workflow system [4].

Sensor data and workflow systems describe properties and location of physical objects at specific points in time with potential imprecision. The basic idea is combining sensor data and workflow data to identify types of imprecisions as well as their origin.

Therefore, physical objects effected by the execution of a workflow and observed by sensors allow to correlate workflow executions and sensor observations. In this paper this correlation is investigated. The correlation is based on a fixed set of workflow executions and sensor data. Since physical objects are not uniquely identifiable, the mapping of sensor and workflow data is potentially not unique.



a0: order products
a1: receive products
a2: check quality, measures etc.
a3: forward to other location
a4: store in local warehouse

Figure 1. Example Procurement Workflow

The contribution in this paper is the representation of the correlation as a coding problem in information theory. Therefore, a probabilistic model based on workflow and sensor data model as a basis for information theory measures is provided. Further, sensor infrastructure error types are distinguished and criteria for identifying these errors based on the introduced information theory measures.

The approach (Sect III) is explained along a running example (Sect II). Details on optimizing the mapping of physical objects (Sect IV) and on the criteria for identifying sensor infrastructure errors (Sect V) are discussed next. The effects of correlating the available data on different granularity levels is discussed in Sect VI.

II. Scenario

As a running example the following procurement workflow is used (see Fig 1). The workflow is denoted as a finite state automaton, where states are represented as circles and activities or labeled state transitions are represented as arcs.

The example workflow starts with centrally ordering a set of products (activity $a0$) electronically, which are later on received in activity $a1$. Next, the quality of the received products is checked (activity $a2$). Depending on the requester of a product products are either forwarded to another branch of the company (activity $a3$) or they remain at the receiving branch, where the products have to be stored in the local warehouse (activity $a4$). This procurement workflow

involves moving physical objects in activities $a1$ to $a4$. While physical objects in activities $a1$ to $a3$ move along the same path, the storage of physical objects into the warehouse (also called sorting) distributes the different physical objects potentially over the complete warehouse, thus the objects move along different paths.

The upper part of Fig 2 depicts three instances of the workflow in Fig 1. Activity $a0$ for all instances is not depicted since it does not have an effect on physical objects. Instance $in1$ contains six physical objects $o1$ to $o6$, which are forwarded to another branch, while instance $in2$ and $in3$ contain three and two physical objects respectively, all sorted into the local warehouse. In Fig 2 each instance is represented along a time line. Further, each activity is depicted as a rectangle where the left and right hand side of the rectangle indicates the start and completion time of the activity respectively. For instance, activity $a1$ of instance $in1$ starts at time one and stops at time three. Therefore, all observed sensor readings in this time interval related to physical objects $o1$ to $o6$ can be correlated to activity $a1$ of instance $in1$.

The lower part of Fig 2 depicts six sensors and the observed physical objects per time step. For example sensor $x1$ does not have any observation at time one, but observes objects $o3$, $o4$, $o1$, and $o8$ at time two.

To illustrate the detection of sensor infrastructure errors, the following issues are contained in this example:

- an error in sensor $x1$ at the beginning of the processing: objects $o1$ and $o2$ of activity $a1$ of instance $in1$ are not observed due to a sensor malfunction.
- equal object identifiers ($o1$ and $o2$) are used by sensor $x1$ belonging to different instances and different activities (for instance $in1$ and $in2$ to activities $a1$ and $a2$)
- equal object identifiers ($o1$ and $o2$) observed by sensor $x2$ belonging to different instances and different activities (for instance $in1$ and $in2$ to activity $a2$)
- objects ($o1$,$o2$, and $o7$) of instance $in2$ are sorted in activity $a4$: objects are observed by sensors $x4$,$x5$,and $x6$.
- object $o9$ in instance $in3$ is treated outside the workflow and therefore the object $o9$ is not observed by any sensor

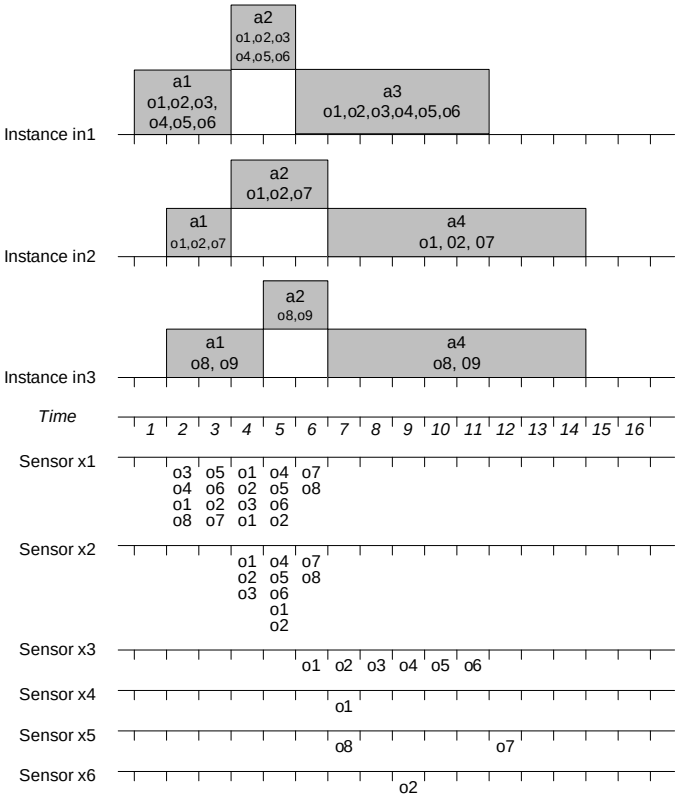


Figure 2. Scenario

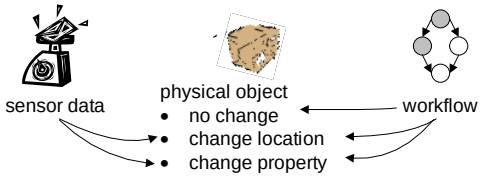


Figure 3. Sensor Data and Workflow Correlation

III. Correlation of Sensor Data and Workflow Systems

The basic idea of correlating workflow systems and sensor data is that both represent a view on physical objects. Executing a workflow means executing activities, which may result in physical objects changing their location or some of their properties. These changes can potentially be detected by an available sensor infrastructure (see Fig 3).

Observing many executions of the same workflow allows to determine probabilities of expected sensor readings if an activity is executed. Thus, if an object is effected by a specific activity, it is expected to be sensed by a subset of sensors. Deviation of the expected sensor readings for a specific activity may be because

- objects are moved different than described in a workflow, or
- objects are not properly observed by the sensor infrastructure due to issues in the sensor infrastructure.

In this paper the focus is on detecting issues in the sensor infrastructure. Changes of physical objects not initiated by a workflow instance are excluded, like e.g. maintenance operations in the warehouse.

A. Scope

The correlation of sensor data and workflow systems is based on physical objects. As a simplification, in this paper no dependencies between physical objects are considered.

For example, sensing a pallet object will not imply sensing several product objects currently standing on this pallet, if the product objects are not observed themselves.

Further, observed physical objects must be distinguishable, i.e., identifiable. However, the physical object IDs do not necessarily be universally unique like e.g. RFID tags, but several physical objects may have the same ID like e.g. bar codes representing a universal product code.

Please be aware that this paper is not addressing syntactic and semantic integration problems [5], although we acknowledge the problem. Further, although sensor data fusion is an important topic, the focus in this paper is on relating sensor data and workflow systems. Therefore, the difficulties of sensor data fusion are not addressed.

To further explain the approach a workflow model and a sensor data model is introduced next.

B. Workflow Model

The workflow schema depicted in Fig 1 is specified as a Finite State Automaton [6]. States are represented as circles and transitions are represented as labeled arrows, where the label represents the activity to be executed in this transition. States with thick lines are called final states, i.e., an execution is successful if it ends in a final state. Each execution starts in the initial state represented by the state with the little arrow pointing at it. An execution of a workflow is called a workflow instance or process, which adheres to the workflow schema. A process execution is a finite sequence of activities connecting the initial state with a final state also known as a trace. With regard to the example, there are only two traces for the workflow schema in Fig 1, i.e., $\langle a0, a1, a2, a3\rangle$ and $\langle a0, a1, a2, a4\rangle$.

To correlate sensor data and processes, several information about the process state is required. In particular, ψ provides the trace of a workflow instance/process in_k. Further, the start and the completion time of an activity within a trace is denote as $\tau_s(a_i)$ and $\tau_c(a_i)$ for the start and completion time of activity a_i in workflow instance/process in_k, i.e., $a_i \in \psi(in_k)$. The IDs of physical objects effected by an activity are denoted as $\theta(a_i)$ providing a multiset[1] of object IDs.

With regard to the example, there are no objects effected by activity $a0$ for all instances in_k, thus $\theta(a0) = \emptyset$. Therefore, activity $a0$ is also not depicted in the upper part of Fig 2. The trace of instance $in1$ is $\psi(in1) = \langle a0, a1, a2, a3\rangle$. For activity $a1$ of instance $in1$ the start time of the activity is $\tau_s(a1) = 1$, the completion time of the activity is $\tau_c(a1) = 3$, and the set of effected objects is $\theta(a1) = \{o1, o2, o3, o4, o5, o6\}$.

[1]A multiset is a set supporting multiple instances of the same element (see Appendix A).

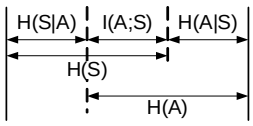


Figure 4. Channel Entropy (taken from [8])

C. Sensor Data Model

A sensor data stream is a discrete-time signal, which can be modeled as a sequence x of sensor readings typically encoded as numbers, where $x[n]$ is the n-th element in the sequence $x = \{x[n]\}$ with $-\infty < n < \infty$ [7]. In this paper, the observations of physical objects made by a sensor at time t results in $x[t]$ being the multiset[1] of observed object IDs. If no objects are observed, the set is empty. Let X be the set of all available sensor data streams.

With regard to the example, there are six sensors available, thus $X = \{x1, x2, x3, x4, x5, x6\}$. The sensor stream $x1$ contains at time $t = 1$ no observations ,i.e. $x1[1] = \emptyset$, while at time $t = 2$ there are four observations, i.e. $x[2] = \{o3, o4, o1, o8\}$.

D. Approach

The approach is based on physical objects effected by the execution of an activity and observed by sensors. In particular, the homogeneity of the correlation of activities and sensor observation based on the effected objects indicates whether the proper operation of the sensor infrastructure.

This problem specification is similar to noisy channel coding theory on a discrete memory less channel (DMC) as discussed in information theory [8]. The more homogeneous the encoding and decoding of messages are the less noisy the communication channel is. In information theory the information entropy is a measure of uncertainty of a stochastic variable. In the context of this paper, the stochastic variables are A representing activities and instances, and S representing sets of sensors. Thus, the *activity entropy* $H(A)$ is based on activities and instances, and the *sensor cluster entropy* $H(S)$ is based on sensor observations. The mutual information $I(A; S)$ (or transinformation) represents the dependency of the stochastic variables for activities and sensor data. Mutual information $I(A; S)$ is based on the notion of entropy (here $H(A)$ and $H(S)$) describing the uncertainty associated with activities and sensor data respectively. The conditional entropy $H(A|S)$ and $H(S|A)$ gives an indication on the uncertainty of the activities based on the sensor data or the other way around. In Fig 4 the dependencies between the different entropies and the mutual information are visualized.

In the following subsections the entropy and mutual information are defined based on the workflow and sensor data model discussed in Sect III-B and III-C.

1) Probability of Activity: The probability function $p(a)$ for an activity a of an instance in_k is the ratio of the number of object IDs associated to activity a, i.e., $|\theta(a)|$, and the number of object IDs of all activities involved in trace $\psi(in_k)$ of all instances $in_1, \dots in_m$.

$$p(a) := \frac{|\theta(a)|}{\sum_{k=1}^{m} \sum_{a_i \in \psi(in_k)} |\theta(a_i)|} \tag{1}$$

The activities in different instances are treated uniquely, although they may be named the same, like e.g. $a1$ in the example is used in the traces of all three instances $in1$, $in2$ and $in3$. The sum of all activity probabilities is one.

The corresponding activity entropy for all instances $in_1, \dots, in_m$ with $A := \{a \in \psi(in_k) | 1 \leq k \leq m\}$ is then defined as [8]

$$H(A) = \sum_{a_i \in A} -p(a_i) * log(p(a_i)) \tag{2}$$

2) Sensor Partition: The objects effected by processes are observed by sensors. Thus, several activities of several instances may contribute to the objects observed by a single sensor. For example, sensor $x1$ in the example observes objects from activities $a1$ and $a2$ of instances $in1$, $in2$, and $in3$. Thus a partition of sensor observations per activity and instance must be defined.

A necessary condition of the partition is that all objects in a partition of sensor x related to activity a must be observed while the activity is executed, i.e. Eq 3 is not empty.

$$\theta(a) \cap \biguplus_{t=\tau_s(a)}^{\tau_c(a)} x[t] \tag{3}$$

A partition can then be defined based on a mapping of sensor readings to activity and instance:

$$\mu(x[t], a) := \begin{cases} \{o \in O | O \subseteq \theta(a) \cap x[t]\} & if \quad \tau_s(a) \leq t \leq \tau_c(a) \\ \emptyset & otherwise \end{cases} \tag{4}$$

with

$$\biguplus_{k=1}^{m} \biguplus_{a \in \psi(in_k)} \mu(x[t], a) \quad = \quad x[t]$$

The partition of activity a of instance in_k is then the union of all mapped object IDs of sensor x during the execution of activity a, i.e.,

$$P_a := \biguplus_{t=\tau_s(a)}^{\tau_c(a)} \mu(x[t], a) \tag{5}$$

The aim is to define a partition which minimizes the errors in the correlation (see Sect IV).

With regard to the example, two partitions for sensor $x1$ derived from Eq 5 are given below:

- $P_{a1,in1} = \{o1, o2, o3, o4, o5, o6\}$,
  $P_{a2,in1} = \{o1, o2, o3, o4, o5, o6\}$,
  $P_{a1,in2} = \{o7\}$, $P_{a2,in2} = \{o1, o2, o7\}$,
  $P_{a1,in3} = \{o8\}$, $P_{a2,in3} = \{o8\}$
- $P_{a1,in1} = \{o3, o4, o5, o6\}$,
  $P_{a2,in1} = \{o1, o2, o3, o4, o5, o6\}$,
  $P_{a1,in2} = \{o1, o2, o7\}$, $P_{a2,in2} = \{o1, o2, o7\}$,
  $P_{a1,in3} = \{o8\}$, $P_{a2,in3} = \{o8\}$

The two partitions deviate in activity $a1$ of instance $in1$ and $in2$, since both activities effect objects $o1$ and $o2$ and the activities overlap in their execution. With regard to these observations, two additional partitions exist for sensor $x1$ varying the assignment of objects $o1$ and $o2$ to activity $a1$ of instance $in1$ and $in2$.

3) Sensor Clustering and Sensor Errors: An object effected by an activity may be observed by several sensors. Thus, the observations made by a set of sensor data streams x_j must be clustered to one observation made by a set of sensors. Therefore, the set of sensor clusters $S := 2^X$ is defined as the powerset of all available sensor data streams X. Based on the mapping function of Eq 4 a cluster s for an activity a of instance in is observing a set of object IDs, i.e.,

$$\bigcap_{x \in s} \biguplus_{t=\tau_s(a)}^{\tau_c(a)} \mu(x[t], a)$$

With regard to the example, sensor $x1$ and $x2$ are clustered for activity $a2$ for all three instances, thus,

- $\{x1, x2\}_{a2,in1} = \{o1, \dots, o6\}$,
- $\{x1, x2\}_{a2,in2} = \{o1, o2, o7\}$ and
- $\{x1, x2\}_{a2,in3} = \{o8\}$.

As can be seen for activity $a2$ of instance $in3$, only object $o8$ is observed, while object $o9$ is not observed. Sensor observation errors are available as an error set $\varepsilon(a) \subseteq E$, where the set E contains the set of all possible errors, i.e. the set of all object IDs of all instances $in_1, \dots, in_m$.

$$E = \bigcup_{k=1}^{m} \bigcup_{a_i \in \psi(in_k)} \theta(a_i)$$

The set of not observed object IDs is $\varepsilon(a) \subseteq E$ for an activity a of an instance in_k is

$$\varepsilon(a_i) = \theta(a_i) \setminus \Big(\biguplus_{x \in X} \biguplus_{t=\tau_s(a_i)}^{\tau_c(a_i)} \mu(x[t], a_i) \Big) \tag{6}$$

With regard to the example, the set E of object IDs is $E = \{o1, \dots, o9\}$ and the set of not observed object IDs $\varepsilon(a_1)$ for activity $a1$ of instance $in3$ is object $o9$, since there is no sensor which observed the object during the execution time of the activity.

4) Probability of Sensor Clusters and Errors: The probability for a sensor cluster $s_j \in S$ is the ratio of the number of objects observed by all sensors in the cluster s_j and the number of all potentially observable objects. Since the sensor infrastructure is considered to be erroneous, the

number of observable objects is taken from the workflow system as it has been used for activity probabilities (see Eq 1).

$$p(s_j) = \sum_{k=1}^{m} \sum_{a_i \in \psi(in_k)} \frac{|\bigcap_{x \in s_j} \biguplus_{t=\tau_s(a_i)}^{\tau_c(a_i)} \mu(x[t], a_i)|}{|\theta(a_i)|} \tag{7}$$

The corresponding sensor cluster entropy for all sensor clusters $s_j \in S$ is defined as

$$H(S) = \sum_{s_j \in S} -p(s_j) * log(p(s_j)) \tag{8}$$

Due to the errors in sensor readings, the sum of all sensor cluster probabilities may be less than 1, i.e.,

$$\sum_{s_j \in S} p(s_j) \leq 1$$

Therefore, the error probability is discussed next. Based on the set of not observed object IDs (see Eq 6) the error probability can be defined similar to the sensor cluster probability. The error probability is defined as the ratio of the sum of not observed object IDs in all activities and the number of all potentially observable objects. For an object ID $o \in E$ the error probability is

$$p(o) = \sum_{k=1}^{m} \sum_{a_i \in \psi(in_k)} \frac{|\{o\} \cap \varepsilon(a_i)|}{|\theta(a_i)|} \tag{9}$$

Due to the construction of the error probability, the sum of error probabilities over all objects in E indicates the error of the sensor readings such that,

$$1 - \sum_{s_j \in S} p(s_j) = \sum_{o \in E} p(o)$$

To make the sensor cluster entropies comparable, the error probabilities should be included, therefore a sensor cluster entropy with error $H^e(S)$ is defined based on the sensor cluster entropy (see Eq 8) as follows [8]

$$H^e(S) = H(S) + \sum_{o \in E} -p(o) * log(p(o)) \tag{10}$$

5) Mutual Information: Based on the partition (see Eq 5) conditional probabilities for sensor readings dependent on an activity can be defined by combining sensor cluster probability (Eq 7) and activity probability (Eq 1).

$$p(s_j|a_i) = \frac{|\bigcap_{x \in s_j} \biguplus_{t=\tau_s(a_i)}^{\tau_c(a_i)} \mu(x[t], a_i)|}{|\theta(a_i)|}$$

Similar to the sensor cluster probability, the error probability (Eq 9) can be expressed as a conditional probability dependent on the activity probability (Eq 1).

$$p(o|a_i) = \frac{|\{o\} \cap \varepsilon(a_i)|}{|\theta(a_i)|}$$

Finally the mutual information can be defined. The mutual information represents the dependency of activities and sensor data. The definition of the mutual information is defined as [8]

$$I(A;S) = \sum_{s_j \in S} \sum_{a_i \in A} p(s_j|a_i)p(a_i)log(\frac{p(s_j|a_i)}{p(s_j)}) \tag{11}$$

Similar to the sensor cluster entropy also the mutual information can be extended by considering errors to provide comparability of the mutual information. The mutual information with errors $I^e(A;S)$ is based on the mutual information as follows

$$I^e(A;S) = I(A;S) + \sum_{o \in E} \sum_{a_i \in A} p(o|a_i)p(a_i)log(\frac{p(o|a_i)}{p(o)}) \tag{12}$$

IV. Partition Optimization

In Sect III-D2 formally a partition of sensor observations has been introduced based on the mapping μ of sensor observations to activities (Eq 4). In the scenario used in this paper (see Sect II), the assignment of objects $o1$ and $o2$ in sensor $x1$ and $x2$ determines the number of possible partitions. The observations on $o1 \in x1[2]$ and $o2 \in x1[3]$ can be associated with activity $a1$ in instance $in1$ or instance $in2$ resulting in four possible partitions. The observations $o2 \in x1[4]$, $o2 \in x1[5]$, and the two observations of $o1 \in x1[4]$ can be assigned to activity $a2$ of instances $in1$ and $in2$. This results again in four possible partitions. However, since there are in total four observations, the four partitions all assign an observation to objects in the activity $a2$ in each instance. Therefore, the partitions are symmetric. The same applies to observations $o1 \in x2[4]$, $o2 \in x2[4]$, $o1 \in x2[5]$, and $o2 \in x2[5]$ which can be assigned to activity $a2$ of instances $in1$ and $in2$. Thus, there are four non symmetric partitions varying the assignment of $o1$ and $o2$ in sensor $x1$ to activity $a1$ in instance $in1$ and $in2$.

The idea is to select a partition with a minimal conditional entropy $H(S|A)$ indicating the uncertainty of the sensor observations based on the activities. To ensure comparability of results, the sensor cluster entropy and the mutual information with errors is used. Since $H^e(S|A) = H^e(S) - I^e(A;S)$ based on Eq 10 and 12 the optimal partition is given by

$$argmin_{\mu} \quad H^e(S) - I^e(A;S)$$

In the scenario, the different measurements are listed in Tab I. The optimal partition is the last one in the table, where the objects $o1$ and $o2$ observed at sensor $x1$ are mapped to activity $a1$ of instance $in1$.

In future research, it will be investigated how to determine the optimal partition without enumerating and inspecting all possible partitions, since the enumeration may result in a combinatorial explosion.

Mapping	$H^e(S)$	$I^e(A;S)$	$H^e(S) - I^e(A;S)$
$o1, o2 \rightarrow in2$	1.8270	1.4434	0.3836
$o1 \rightarrow in1$, $o2 \rightarrow in2$	1.8270	1.4613	0.3657
$o1 \rightarrow in2$, $o2 \rightarrow in1$	1.8270	1.4613	0.3657
$o1, o2 \rightarrow in1$	1.8270	1.5012	0.3258

Table I
NON-SYMMETRIC PARTITIONS OF OBJECTS $o1$ AND $o2$ MAPPED TO INSTANCE $in1$ AND $in2$

V. SENSOR INFRASTRUCTURE ERRORS

Based on the introduced entropy and mutual information measures (see Sect III-D), criteria for assessing a correlation of workflow and sensor data are provided.

The sensor infrastructure is working without error, if the activity entropy and the mutual information have the same value. Therefore, if there is an object of any activity, which is not observed by any sensor, then the mutual information is smaller than the activity entropy.

In case of a correct working sensor infrastructure the sorting and picking can be distinguished from a homogeneous execution. The execution is **homogeneous** if the sensor cluster entropy equals the mutual information. Intuitively this means that all objects of an activity are observed by the same sensor cluster, thus, all objects follow the same path. In case of **sorting and picking** at an activity, objects are observed by varying sensor clusters.

In case of a not correct working sensor infrastructure, the occurrence of ignored objects can be distinguished from sensing errors. **Ignored objects**, means that there are objects belonging to activities, which are not observed by any sensor cluster. Therefore the mutual information with and without error is the same, i.e., $I^e(A;S) = I(A;S)$. However, if objects belonging go activities are sometimes observed and sometimes not than this a **sensing error** is detected and the mutual information with error is greater than the mutual information, i.e., $I^e(A;S) > I(A;S)$. The proofs for the statements made in this section are elaborated in the following subsections. A decision tree of the above statements is depicted in Fig 5.

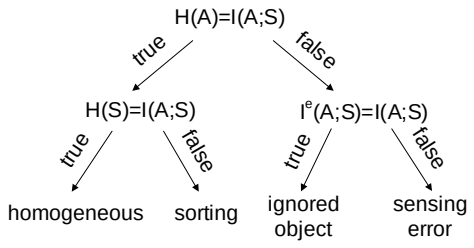


Figure 5. Decision tree for sensor infrastructure errors

A. Complete observation of objects

All objects associated to activities are observed in the sensor data if the activity entropy equals the mutual information.

Lemma 1: $H(A) = I(A;S)$ iff for all sensors s_j, all objects of activity a_i with $p(a_i) > 0$ are observed by sensor s_j.

Proof: Based on the lemma $-H(A) + I(A;S) = 0$. With Eq 2 and the re-written Eq 11 the difference can be written as

$$\sum_{a_i \in A} p(a_i) * log(p(a_i)) + \sum_{s_j \in S} \sum_{a_i \in A} p(a_i|s_j) p(s_j) log(\frac{p(a_i|s_j)}{p(a_i)}) = 0$$

which can be re-written into

$$\sum_{a_i \in A} \Big(\begin{array}{l} p(a_i) * log(p(a_i)) + \\ \sum_{s_j \in S} p(a_i|s_j) p(s_j) log(p(a_i|s_j) - \\ log(p(a_i)) * \sum_{s_j \in S} p(a_i|s_j) p(s_j) \Big) = 0 \end{array}$$

by combining the sums over the activities and separating the division in the logarithm into two terms. Since $\sum_{s_j \in S} p(a_i|s_j) p(s_j) = p(a_i)$ based on Bayes, the first and third term compensate each other, thus

$$\sum_{a_i \in A} \sum_{s_j \in S} p(a_i|s_j) p(s_j) log(p(a_i|s_j)) = 0$$

If $p(s_j) = 0$ then the sensor is not recording any objects and therefore can be neglected. If $p(s_j) > 0$ then an object is recorded, which can be associated with an activity a_i. Based on a condition of the lemma $p(a_i) > 0$, also the conditional probability is greater than zero, i.e., $p(a_i|s_j) > 0$. Thus, the above expression can only be zero if $p(a_i|s_j) = 1$. The conditional probability is one, i.e., $p(a_i|s_j) = 1$ iff all objects observed by sensor s_j belong to activity a_i. This proofs the lemma. ■

B. Homogeneity of Observations

All objects associated to activities are observed in the sensor data if the sensor cluster entropy equals the mutual information.

Lemma 2: $H(S) = I(A;S)$ iff for all activities a_i, all objects of activity a_i with $p(a_i) > 0$ are observed by sensor s_j.

The proof goes along the lines of the proof in the previous section.

Proof: Based on the lemma $-H(A) + I(A;S) = 0$. With Eq 8 and Eq 11 the difference can be written as

$$\sum_{s_j \in S} p(s_j) * log(p(s_j)) + \sum_{s_j \in S} \sum_{a_i \in A} p(s_j|a_i) p(a_i) log(\frac{p(s_j|a_i)}{p(s_j)}) = 0$$

which can be re-written into

$$\sum_{s_j \in S} \Big(\begin{array}{l} p(s_j) * log(p(s_j)) + \\ \sum_{a_i \in A} p(s_j|a_i) p(a_i) log(p(s_j|a_i)) - \\ log(p(s_j)) * \sum_{a_i \in A} p(s_j|a_i) p(a_i) \Big) = 0 \end{array}$$

by combining the sums over the activities and separating the division in the logarithm into two terms. Since $\sum_{a_i \in A} p(s_j|a_i)p(a_i) = p(s_j)$ based on Bayes, the first and third term compensate each other, thus

$$\sum_{s_j \in S} \sum_{a_i \in A} p(s_j|a_i)p(a_i)log(p(s_j|a_i)) = 0$$

If $p(a_i) = 0$ then the activity does not contain any recorded objects and therefore can be neglected. If $p(a_i) > 0$ then an activity does contain a recorded object, which can be associated with sensor s_j. Thus, the above expression can only be zero if $p(s_j|a_i) = 1$. The conditional probability is one, i.e., $p(s_j|a_i) = 1$ iff all objects of activity a_i are observed by sensor s_j. This proofs the lemma. ■

C. Sensing Error

All objects associated to activities are observed in the sensor data if the sensor cluster entropy equals the mutual information.

Lemma 3: $I^e(A;S) = I(A;S)$ iff there are no erroneous observations or a set of objects is not observed for all activities a_i with $p(a_i) > 0$.

Proof: Based on the lemma, $I^e(A;S) - I(A;S) = 0$. With Eq 12 and Eq 11 the equation can be re-written as

$$\sum_{o \in E} \sum_{a_i \in A} p(o|a_i)p(a_i)log(\frac{p(o|a_i)}{p(o)}) = 0$$

. This equation is fulfilled if either the conditional probability of an erroneous reading is zero, i.e., $p(o|a_i) = 0$ or the conditional probability of an erroneous reading equals the probability of an erroneous reading, i.e., $p(o|a_i) = p(o)$.

The case of $p(o|a_i) = 0$ means that all objects are observed per activity, thus, does not have any erroneous objects.

For the second part, it has to be shown that $p(o|a) = p(o)$ for all $a \in A$ where $p(o|a) > 0$. Since

$$p(o) = \sum_{a_i \in A} p(o|a_i)p(a_i)$$

, the condition can be rephrased as

$$p(o|a) = \sum_{a_i \in A} p(o|a_i)p(a_i)$$

. Since $p(o|a) > 0$ the equation is dived by the term resulting in

$$\sum_{a_i \in A \setminus \{a\}} \frac{p(o|a_i)}{p(o|a)} p(a_i) + p(a) = 1$$

. With $\sum_{a_i \in A} = 1$ and since the equation must hold for all a the equation can only be fulfilled if

$$\frac{p(o|a_i)}{p(o|a)} = 1$$

for all a_i. This means that all probabilities for erroneous readings is equal for all activities.

Mapping	$\sum_{k=1}^{3} H^e(S_k) - I^e(A_k;S_k)$
$o1, o2 \rightarrow in2$	1.3486
$o1 \rightarrow in1$, $o2 \rightarrow in2$	1.4218
$o1 \rightarrow in2$, $o2 \rightarrow in1$	1.4218
$o1, o2 \rightarrow in1$	1.4256

Table II
NON-SYMMETRIC PARTITIONS OF OBJECTS $o1$ AND $o2$ MAPPED TO INSTANCE $in1$ AND $in2$ PERFORMED ON THE INSTANCE LEVEL

As a consequence of the second part, it can be seen that all $p(o|a_i)$ must have the same value which can be either null in case of no errors or a specific probability in case of errors stemming from ignored objects. ■

VI. GRANULARITY

The scenario contains several issues as discussed in Sect II. Based on the current values of entropy and mutual information, the various sensing errors overlap. This makes it impossible to decide whether an object has been ignored, whether there is sorting or picking, or whether there is indeed a sensor infrastructure error.

As a consequence, the information of workflow system and sensor infrastructure must be investigated in smaller portions to separate different issues. Obvious ways to partition the available information is by investigating instances separately, or by partitions of sensors. By reducing the granularity of the data set, on the one hand side issues potentially get separated and therefore identifiable. On the other hand side, if the granularity is getting to fine grained, like e.g. single sensors or a single activity of an instance, then there is not sufficient mutual information to conclude anything.

With regard to the running scenario, investigating the data on instance level separates the issues of the scenario. The formulas for deriving the individual measures on instance level are the same as presented in Sect III-D except that there is no aggregation over instances anymore.

Due to different granularity and the various effects on the entropy and mutual information, the partition optimization and the sensor infrastructure error identification have to be performed on the instance based data.

A. Partition Optimization

Partition optimization as introduced in Sect IV aims at minimizing the uncertainty of sensor observations based on activities. The sum of the distance of sensor cluster entropy and mutual information over all instances is depicted in Tab II. The assignment of objects $o1$ and $o2$ to activity $a1$ of instance $in2$ produces the minimal result. This is a change to the initial assessment in Sect IV. In the following this partition is used.

Instance	$H(A)$	$I(A;S)$	$H(S)$	$I^e(A;S)$	$H^e(S)$
$in1$	1.0986	0.9765	1.0666	1.0986	1.3878
$in2$	1.0986	1.0986	1.4648	1.0986	1.4648
$in3$	1.0986	0.5493	0.8959	0.5493	1.2425
$in3$ w/o $o9$	1.0986	1.0986	1.0986	1.0986	1.0986

Table III
ENTROPY AND MUTUAL INFORMATION PER INSTANCE

B. Sensor Infrastructure Errors

The entropy and mutual information measurements per instance are contained in Tab III. The criteria introduced in Sect V are used for the identification of errors. Considering the first three rows in Tab III representing the measures for the three instances, there is no instance working correctly, but all have sensing errors.

Instance $in2$ fulfills the conditions for **sorting and picking**, i.e., $H(A) = I(A;S)$ and $H(S) > I(A;S)$. This is indeed the case as described in Sect II. In the scenario the objects $o1$,$o2$, and $o7$ effected by activity $a4$ of instance $in2$ are sensed by sensors $x4[7]$, $x6[9]$ and $x5[12]$ respectively (see Tab III second row).

Instance $in3$ fulfills the condition for **ignored objects**, i.e., $H(A) > I(A;S)$ and $I^e(A;S) = I(A;S)$. Checking the scenario shows that object $o9$ is never sensed, thus although it occurs in all activities of instance $in3$. Therefore, it seems the object is treated outside the workflow (see Tab III third row).

If we exclude object $o9$ from the process information of instance $in3$, then the entropy and mutual information values depicted in the last row of Tab III are calculated. This row fulfills the criteria for a **homogeneous** working sensor infrastructure for this instance, i.e., $H(A) = I(A;S)$ and $H(S) = I(A;S)$. In this case, the entropy and mutual information of instance $in1$ and $in2$ are the same as in the first two rows of Tab III.

Finally, instance $in1$ indicates a **sensing error**, i.e., $H(A) = I(A;S)$ and $I^e(A;S) > I(A;S)$. Due to the used mapping of assigning objects $o1$ and $o2$ to instance $in2$ rather than instance $in1$ the objects $o1$ and $o2$ used in activity $a1$ of instance $in1$ do not have matching sensor observations, thus, it is an issue with the sensing infrastructure (see Tab III first row).

VII. RELATED WORK

Up to my knowledge, the idea of loosely coupling sensor data and workflow states has not been discussed in literature before. There is however, quite some literature on correlating different models based on their overlap describing a system. Examples are correlating business process models and coordination models at design time [9] or at run time [10], or the monitoring of Web Service compositions using logging information [11]. However, sensor data and workflow systems are indirectly correlated via the state of the physical world, while the afore mentioned approaches do the correlation directly.

The mapping function between sensor data and workflow state has some similarities with data integration approaches like e.g. [5] for homogeneous models. In particular, the approach of probabilistic data integration could be a good source of inspiration [12].

Please be aware that although dealing with sensor data scientific workflows are very different from business workflows as discussed in this paper. Scientific workflows are focusing on processing data often coordinated by data [13], [14] while business workflows have many instances of the same workflow schema.

In [15] the authors propose a model of physical objects to identify cardinality, frequency and duration properties of physical objects related to a business workflow. Such a model could be beneficial in the context of this work. However, in the current paper, this additional explication of the physical objects in terms of a model is not required.

Mutual information and information entropy have been utilized in several application domains. An application close to the one in this paper is the assessment or selection of alternative models. This is e.g. done in [16] by optimizing mutual information related to a Bayesian learning algorithms. Model selection based on changes in entropy are proposed e.g. in [17]. Both cases deviate from the work presented here since the selections are made of model alternatives of the same model type, while in this paper, models of different types are correlated, namely sensor data and workflow data.

VIII. CONCLUSION

The correlation of sensor data and business workflow data based on mutual information and information entropy seems to be a good approach to classify the correlation. This classification can be used to identify changes in the sensor infrastructure or errors in observing physical objects. They actually provide a metric for the health of the sensor infrastructure and the business workflow.

In future work, more work will be invested in identifying business workflow errors as well as the development of guidelines for selecting the appropriate granularity level of inspecting the data. Further, the current results will be applied in a case study.

REFERENCES

[1] R. Jedermann, C. Behrens, D. Westphal, and W. Lang, "Applying autonomous sensor systems in logistics–combining sensor networks, rfids and software agents," *Sensors and Actuators A: Physical*, pp. 370 – 375, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/B6THG-4JG5FBT-4/2/2dd9c816f409137409e604c48b68db05

[2] M. Wieland, U.-P. Käppeler, P. Levi, F. Leymann, and D. Nicklas, "Towards integration of uncertain sensor data into context-aware workflows," in *GI Jahrestagung*, 2009, pp. 2029–2040.

[3] P. Soffer, "Mirror, mirror on the wall, can i count on you at all? exploring data inaccuracy in business processes," in *Proceedings of BPMDS*, 2010.

[4] R. Lenz and M. Reichert, "It support for healthcare processes - premises, challenges, perspectives," *Data Knowl. Eng.*, vol. 61, no. 1, pp. 39–58, 2007.

[5] E. Rahm and P. A. Bernstein, "A survey of approaches to automatic schema matching," *VLDB J*, vol. 10, no. 4, pp. 334–350, 2001. [Online]. Available: http://link.springer.de/link/service/journals/00778/bibs/1010004/10100334.htm

[6] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*. Addison Wesley, 2001.

[7] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, ser. Prentice Hall Signal Processing Series. Englewood Cliffs, NJ, USA: Prentice Hall, 1989.

[8] H. Tzschach and G. Haßlinger, *Codes für den störungssicheren Datentransfer*. Oldenburg, 1993.

[9] Z. Zlatev and A. Wombacher, "Consistency between e^3-value models and activity diagrams in a multi-perspective development method," in *OTM Conferences (1)*, 2005, pp. 520–538. [Online]. Available: http://dx.doi.org/10.1007/11575771_33

[10] L. Bodenstaff, A. Wombacher, M. U. Reichert, and R. Wieringa, "Monitoring Collaboration from a Value Perspective," in *Intl Conf on Digital Ecosystems and Technologies*, 2007.

[11] L. Bodenstaff, A. Wombacher, M. Reichert, and M. C. Jaeger, "Monitoring dependencies for SLAs: The mode4SLA approach," in *IEEE SCC*, 2008, pp. 21–29. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/SCC.2008.120

[12] M. van Keulen, A. de Keijzer, and W. Alink, "A probabilistic XML approach to data integration," in *ICDE*. IEEE Computer Society, 2005, pp. 459–470. [Online]. Available: http://csdl.computer.org/comp/proceedings/icde/2005/2285/00/22850459abs.htm

[13] W. Tan, P. Missier, R. K. Madduri, and I. T. Foster, "Building scientific workflow with taverna and BPEL: A comparative study in cagrid," in *ICSOC Workshops*, 2008, pp. 118–129. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-01247-1

[14] B. Ludäscher, M. Weske, T. M. McPhillips, and S. Bowers, "Scientific workflows: Business as usual?" in *BPM*. Springer, 2009, pp. 31–47. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03848-8

[15] R. Wieringa, V. Pijpers, L. Bodenstaff, and J. Gordijn, "Value-driven coordination process design using physical delivery models," in *ER*, ser. Lecture Notes in Computer Science, Q. Li, S. Spaccapietra, E. S. K. Yu, and A. Olivé, Eds., vol. 5231. Springer, 2008, pp. 216–231. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-87877-3_17

[16] Y. Zeng and P. Doshi, "Model identification in interactive influence diagrams using mutual information," *Web Intelli. and Agent Sys.*, vol. 8, no. 3, pp. 313–327, 2010.

[17] I. Vatcheva, H. de Jong, O. Bernard, and N. J. I. Mars, "Experiment selection for the discrimination of semi-quantitative models of dynamical systems," *Artificial Intelligence*, vol. 170, no. 4–5, pp. 472–506, 2006.

APPENDIX

A multiset $\mathcal{A}$ consists of a set of elements A and a function $m : A \to N$ called multiplicity assigning each element a natural number of occurrences. If the multiplicity of an element is zero then the element is not in the multiset anymore.

Intersection of two multisets $\mathcal{A} \cap \mathcal{B}$ is the minimum multiplicity of each element contained in the multisets, i.e. $\mathcal{A} \cap \mathcal{B}$ consists of the intersection of the sets of elements $A \cap B$ and the minimum multiplicity for each element of the intersection of the sets of elements,
$\forall e \in A \cap B . m_{A \cap B}(e) := min(m_A(a), m_B(e))$.

The **union** of two multisets $\mathcal{A} \cup \mathcal{B}$ is the maximum multiplicity of each element contained in the multisets,
$\forall e \in A \cup B . m_{A \cup B}(e) := max(m_A(a), m_B(e))$.

The **multiset sum** of two $\mathcal{A} \uplus \mathcal{B}$ is the sum of the multiplicity of each element contained in the multisets,
$\forall e \in A \uplus B . m_{A \uplus B}(e) := m_A(a) + m_B(e)$.

The **multiset difference** of two multisets $\mathcal{A} \setminus \mathcal{B}$ is the maximum of zero and the difference of the multiplicity of $\mathcal{A}$ and $\mathcal{A}$ per element, i.e., $\forall e \in A . m_{A \setminus B} := max(0, m_A(a) - m_B(e))$.

The **cardinality** of a multiset $|\mathcal{A}|$ is the sum of the multiplicity of all elements in A, i.e. $|\mathcal{A}| = \sum_{e \in A} m(e)$.