1 Application of 3P-TES to Completely Different Source of Sensor Data

1.1 Farmer's Data - One non-static/mobile sensor

The second example we will use to illustrate the Total Error Framework for Sensor Data will be on a non-static sensor. Punt, Snijkers, and de Broe (2019) have explored data concerning smart precision farming by an innovative farmer in the Netherlands, in order to assess its opportunities for NSIs. By employing sensors, actions can be tailored specifically for the different needs of different sections of the field. While the farmer makes use of both static and non-static sensor systems, we will focus on a non-static sensor employed during the collection season of potatoes in fall. One of the sensors employed in this situation assesses the weight of the potatoes.

Acquisition layer - Representation

In this example, the *Target Object* would be the potatoes on a specific section of the field. We are not (exactly) dealing with one large network of sensors, but one sensor moving over the whole field. Therefore, each section of the potato field, where the sensor makes a measurement, is regarded as an object during phase 0, in the acquisition layer. The *Accessible Object* refers to those collected potatoes in section A, which could be recognized by the sensor. Accordingly, the *Object-Frame Error* therefore describes the differences between the collected potatoes in section A, and what the sensor assumes to be the collected potatoes in section A. Imagine, that the potatoes are still covered by a lot of dirt, if that is weighed by the sensor, overcoverage is present. Dirt is included, which should not be a part of the collected potatoes in section A.

The *Accessed Object* therefore refers to the actually weighed potatoes, not only those theoretically accessible to the sensor. If the sensor is installed in a way that some potatoes pass the weighing-ares/-platform too fast, the sensor would not be able to access and assess their weight. *Object-Selection Error* over time. Again, the Object-Selection-Error can be described as an issue relating more strongly to the employed sensor itself, while the previous Object-Frame Error relates mostly to issues with the object itself.

Discrepancies between the Accessed Object - the accessed potatoes in section A - and the *Observed Object* are characterized by missingness or redundancy. If we are missing an observation on potatoes in section C, this is characterized by the notion of *Object-Missing & -Redundancy*. As only one sensor is employed, we are likely to have only one observation per object. This means, that unlike with the example of the water data, all missing observations can be characterized by the notion of Object-Missing & -Redundancy. While the error source may still lie in the measurement-line of processes, this affects the representation of the potatoes in section A.

Acquisition layer - Measurement

In this example, the **Target Aspect** is the weight of the potatoes in section A. As the farmer would like to get this information immediately after collection, a sensor is installed in the machine driving over the fields. This sensor measures a *Target Measure*, which is supposed to represent the Target Aspect as well as possible. In our case, the weight of the potatoes is both the aspect that we intend to employ as well as the measure measured by the sensor. However, imagine the farmer used a sensor that did not weigh the potatoes, but counted them as the passed by. The Target Aspect would not be accurately represented by the Target Variable, a *Validity Error* would have been committed without proper modelling of potato weight based on the count data.

The sensor is employed to assess said variable. Whatever the sensor senses in section A, therefore characterizes the *Collected Measure*. Differences fall under the *Measurement Error*. If the sensor does not accurately measure the weight of the potato, but only within a certain range (with noise), the Target and Collected Measure are not exactly the same. It is likely, that the weight sensor over the years becomes less accurate. Due to wear-and-tear in a part of the sensor-unit, the platform does not return back up to a state giving measurements of zero, when no potatoes are on it. Instead, continuous positive measurements are made. This is similar to the notion of sensor-drift and changes the access of the sensor to the object, increasing the size of the Measurement Error.

After the sensor assessed the weight, this information needs to be recorded, resulting in a *Observed Measure*. This is necessary, so computer (and ultimately humans) are able to process and use the information by the sensor. Whenever the information is not accurately represented in the binary code, a *Recording Error* is committed. For example, it may be possible that the sensor is faulty after a software update, and only returns zero-measurements. Another instance of Recording Error concerns the storage of measurements. As our machine is moving through the sections, the measurements will need to be saved either in a local chip on the machine, or in a remote database using some sort of wireless gateway. Should this gateway be offline, or the storage unit be overfilled, observations would go missing.

Processing layer - Representation

As phase 1 is equivalent to the processing layer, the target of interest no longer lies at the object, but at the network. While there is no network of sensors in this specific example of a potato field, the difference between object and network remains crucial. One sensor does all measurements of all objects - all sections on the field. Therefore, the entire potato field is seen as the *Target Network* in this case, a network of sections creating the entirety of the potato field. The *Accessible Network* subsequently represents all those sections, that should be accessed by the sensor. Before such a machine rolls over the fields, a plan or map has to be exist. This map or plan has to contain all sections of the potato field, this may be done in form of a shape-file containing describing all locations of the potato field. Errors in such a file, where sections may be missing or erroneously included constitute a *Network-Frame Error*, and can be described via notions of over- and under-coverage.

The *Accessed Network* subsequently represents the network, that is actually monitored by the sensor(s). This is best imagined by using a second map or plan, which the machine is programmed to follow (autonomously or by a driver). This plan of how the machine moves, and therefore allows the sensor to measure the individual sections and entire network, has to align with the map of the potato field. If these do not align, the sensor would not be able to cover the entire field. A *Network-Selection Error* would have been committed.

The *Observed Network* represents the information we have on the full potato field. Without any missing observations on sections in the field, the Observed and the Accessed Network would be exactly the same. Accodingly, the notion of *Network-Missing & -Redundancy* describes situations in which observations on entire sections of the field are missing. Due to the nature of the data-collection process with a single sensor, the Object-Missing & -Redundancy is equivalent, as we only have single observations on the section of the field anyway.

Processing layer - Measurement

After (or during) storage, observations are most likely put into relation with other measurement points. The data point alone of section A may not be interesting, but all data points of all sections of the field gives more meaning to each measurement. Often, relational databases are employed, where the continuous stream of data is put into relation with each data-point and other para-data, possible even for long-term storage. However, it may happen that due to an error in the algorithm, data points from section A are stored in relation to section B. This would constitute a *Matching Error*, the measurement was not properly matched. The result of such processes (no matter whether errors are contained or not) is referred to as *Related Measures*.

After finding their way in such a relational data base, measurements often need editing before they can be put to actual use. The format of specific measurements may not be correct, missing data may limit the number of data points or the measurements themselves are rather imprecise, due to an added layer of noise. By editing data, imputing data, or filtering data, errors can be introduced. We might misunderstand the format of the data-points and alter the information. Or the imputation procedure we employed for the missing section is not able to deal with the selectivity of the missing data appropriately. These issues found in the *Edited Measures* are described under the *Processing Error*.

Processing layer - Representation - Integrity

A high level of data integrity, as discussed in the section on water data, should be ensured. For secondary data use through external data analysts, meta- and para-data need to be well-documented. For somebody that did not collect the data themselves, it is not only crucial to be aware of the means through which the data was collected. Detailed documentation on objects and network, just as imputation and processing procedures have to be supplied in a read-me.txt or other additional text or data files. Otherwise, an *Integrity Error* is committed.

References

Punt, T., Snijkers, G., & de Broe, S. (2019, October). Exploring precision farming data: a valuable new data source? a first orientation. In *Workshop on statistical data collection 'new sources and new technologies'.*