Hsieh et al. has recently published a paper on the counterfactual generation of process data. They call their framework DiCE4EL and shares many ideas with our framework. Therefore, we want to highlight the key differences and similarities.

Framework Differences

In their approach they attempt to solve various issues that we have also highlighted in **??**. Furthermore, they do so by incrementally generating the model in a sequential order. However, unlike Oberst and Sontag, whose solution creates counterfactuals for every step in the sequence, Hsieh et al. focus on critical decision points they call milestones.

To gain a better understanding, it is important to briefly outline the event log the authors use. It was taken from a Dutch bank which processes loan applications in which customers request a certain amount of money. The activities relate to either application states or manual work activities. The application states are basically super tasks generated by a machine and manual work activities produced by humans their substates. Now, to understand why the milestone approach works, requires to know that an application loan process will only be rejected after an application state events occured. Not, if a human subtask occured.

Thus, we can safely focus on these application state events to evaluate what is necessary to avoid an undesired outcome. A crucial distinction to our proposed framework, is their crucial dependence on knowledge about the true process model as displayed in this section. Our framework does not leverage structural information about the true process model in question. We believe this is the core contribution in comparison to their approach.

However, similarities between both frameworks do exist. Mainly, our approach also relies on prediction scores of the model we attempt to explain. Similar to Hsieh et al., we incorporate these scores into our quality measure.

Viability

The next difference stems from how DiCE4EL operationalises the quality assessment of their counterfactuals and on what bases they optimize their algorithm.