A behavioural model of a system is useful to help engineers understand how a system is functioning and audit it. However, creating such a model by hand is a long and error-prone task. Model learning methods are methods that infer a behavioural model of a system.

There exist two types of model learning methods:

- **Active methods** [1], which query directly the system or the user to obtain observations, used to build or improve a model. This type of method does not work on every system, like uncontrollable systems or systems that cannot be reset like IoT systems.

- **Passive methods** [2], which deduce a model from data extracted from the system like for example execution traces. These methods can return incomplete models, and huge models difficult to understand.

IoT systems are generally composed of many reusable components, and modeling these components separately can lead to smaller and more understandable models.

COnfECt (Correlate Extract Compose) is a passive model learning method that generates model of each component of the system from execution traces of the system. A more detailed presentation of this work can be found in [3].

The **COnfECt Method**

We suppose that the events of the log have the following form:

```plaintext
<label>(parm1, param2, …).
```

The COnfECt method is separated into 4 steps: Trace Extraction, LTS Generation, LTS Synchronization, and state merging.

**STEP 1: Trace Extraction:**
In this step, we try to separate the events that come from different components. For that we use a correlation coefficient, used to determine if two events in a trace come from the same component. This coefficient is defined by the user and depends on the system, for example, it can be based on frequency of the events. We use this coefficient to separate each trace into sequences of consecutive events with a high correlation. Then, we remove of the trace each sequence that come from an other component than the first event of the trace, and replace it by synchronization event `call_Ci` and `return_Ci`.

**STEP 2: LTS Generation:**
In this step, we generate for each the trace Ti, a LTS Ci with only one path that correspond of the trace, where each event is modeled by a transition.

**STEP 3: LTSs Synchronization:**
Different synchronization strategies are proposed for our models. Here only the weak strategy is showed, where we merge the similar LTSs that come from the same component, with the help of clustering technique and generalize the models by letting a component call an other many times in a row (modelled by a loop `call_Ci return_Ci`).

**STEP 4: State Merging:**
In this step we merge the equivalent states with the help of kTail[2] with k=2, that merges states that have the same future of length k.

**Conclusion**

The COnfECt method can generate the model of the different component of the system. These models can be used to analyze the whole system, with the help of the synchronization events `call_Ci` and `return_Ci`, or to analyze a specific component of the system by hiding these events.

In term of perspective, we plan to use these models, to verify some security property.

**Bibliography**