The previous chapter focused on the use of alignments to analyze the data perspective of processes. This chapter presents a decomposition approach of that technique to check conformance of data-aware processes. In particular, the chapter presents a valid decomposition of data-aware models, and a decomposition strategy based on single entry single exit components.
17.1 Introduction

In the previous chapter we introduced the concepts of conformance and alignments for data-aware processes, i.e., processes with both control-flow and data perspectives combined. Checking conformance on data-aware processes is a time-consuming task. In this chapter we propose a decomposition version of the data-aware conformance checking in order to reduce the computation time and improve the understanding of the conformance errors, similar to the one presented in Chapter 13 for control-flow models.

17.2 Valid Decomposition of Data-aware Models

In Chapter 13 a valid decomposition [9] is presented in terms of Petri nets: the overall system net $SN$ is decomposed into a collection of subnets $\{SN_1, SN_2, \ldots, SN_n\}$ such that the union of these subnets yields the original system net. A decomposition is valid if the subnets “agree” on the original labeling function (i.e., the same transition always has the same label), each place resides in just one subnet, and also each invisible transition resides in just one subnet. Moreover, if there are multiple transitions with the same label, they should reside in the same subnet.

As shown in Chapter 12, these observations imply that conformance checking can be decomposed. Any trace that fits the overall process model can be decomposed into smaller traces that fit the individual model fragments. Moreover, if the smaller traces fit the individual fragments, then they can be composed into a trace that fits into the overall process model. This result is the basis for decomposing process mining problems.

In this chapter, the definition of valid decomposition is extended to cover Petri nets with data.

Definition 17.1 (Valid Decomposition for Petri nets with Data) Let $DPN \in \mathcal{U}_{DPN}$ be a Petri net with Data. $D = \{DPN_1, DPN_2, \ldots, DPN_n\} \subseteq \mathcal{U}_{DPN}$ is a valid decomposition if and only if:

- for all $1 \leq i \leq n$: $DPN_i = (SN_i, V_i, val_i, init_i, read_i, write_i, guard_i)$ is a Petri net with Data, $SN_i = (PN_i, M_{init}, M_{final}) \subseteq \mathcal{U}_{SN}$ is a system net, and $PN_i = (P_i, T_i, F_i, l_i)$ is a labeled Petri net,
- $D' = \{SN_1, SN_2, \ldots, SN_n\} \subseteq \mathcal{U}_{SN}$ is a valid decomposition of $\bigcup_{1 \leq i \leq n} SN_i$,
- $V_i \cap V_j = \emptyset$ for $1 \leq i < j \leq n$,
- $DPN = \bigcup_{1 \leq i \leq n} DPN_i$.

$\mathcal{D}(DPN)$ is the set of all valid decompositions of $DPN$.

Each variable appears in precisely one of the subnets. Therefore, $V_i \cap V_j = \emptyset$ implies that there cannot be two fragments that read and or write the same data variables: $\bigcup_{t \in T_i} read_i(t) \cup write_i(t) \cap \bigcup_{t \in T_j} read_j(t) \cup write_j(t) = \emptyset$ for $1 \leq i < j \leq n$. Moreover, two guards in different fragments cannot refer to the same variable. If