

Product Based Workflow Support: Dynamic Workflow Execution

Irene Vanderfeesten, Hajo A. Reijers, and Wil M.P. van der Aalst

Technische Universiteit Eindhoven, Department of Technology Management,
PO Box 513, 5600 MB Eindhoven, The Netherlands
{i.t.p.vanderfeesten,h.a.reijers,w.m.p.v.d.aalst}@tue.nl

Abstract. Product Based Workflow Design (PBWD) is a successful new approach to workflow process support. A description of the product, the Product Data Model (PDM), is central to this approach. While other research so far has focused on deriving a process model from the PDM, this paper presents a way to directly execute the PDM. This leads to a more dynamic and flexible support for the workflow process.

Keywords: Workflow Management, Product Data Model, Process Modeling, Process Execution Strategies.

1 Introduction

Product Based Workflow Design (PBWD) [1,4,5] is a successful new approach to workflow process design in which a description of the workflow product is central. So far, PBWD research has mainly focused on generating process models from a product structure, either manually or automatically. The manual derivation of a process model has turned out to be very time consuming [4]. Experiences with the automatic generation of process models [7] triggered a new idea to provide flexible and dynamic support for process execution *directly* on the basis of the product structure, i.e. without first deriving a process model that describes the desirable flow of work. We will refer to this concept as Product Based Workflow Support (PBWS).

2 Product Based Workflow Support

The product of a workflow process is usually an *informational* product, e.g. the decision on an insurance claim or the allocation of a subsidy. The structure of the workflow product can be described by a tree-like structure similar to a Bill-of-Material from manufacturing [2]. Such a description of a workflow product is called a Product Data Model (PDM). Figure 1(a) shows a very small example of a PDM. Because of space limitations we refer to [7] for the complete explanation of this example.

In general, a PDM consists of a number of data elements (depicted as circles) that are linked to each other through operations (depicted as arcs). Each

operation can have one or more input data elements and produces exactly one output data element. The operation on the input elements can be e.g. a calculation, an assessment by a human, or a rule-based decision to determine the output element. An operation is executable when all of its input elements are available. Moreover, several operations can have the *same* output element while having a different set of input elements. These operations represent alternative ways to produce the output product. Finally, operations can have a number of attributes such as the execution cost, processing time, failure probability and execution conditions.

The basic idea of PBWS is that dynamically, during each step of the process execution, all data elements are determined that are available for a case. At run time, it can then be decided what would be the most proper next step in the execution of the process, considering the information available for the specific case, the underlying product specification, and the desired performance.

2.1 Runtime Execution of a PDM

When the workflow process is executed for a particular case some data elements are initially provided by the client or can be retrieved from other systems. Based on these available data elements, new information is produced step-by-step by executing enabled operations. Figure 1 illustrates how the runtime execution of our example PDM works. Suppose that at the start of the process input data elements B , E , and F are available (see Figure 1(b)). The operations that are now enabled for execution are $Op1$ and $Op2$, since all of their input elements are available (Figure 1(c)). Operation $Op3$ is not executable because data element C is not available yet and $Op4$ is not executable since D is not present. Now, we have to choose which of the two executable operations ($Op1$, $Op2$) we select. Suppose we select $Op1$. Then, data element C is produced (Figure 1(d)). The executable operations are calculated again: $Op2$ and $Op3$. And one of those operations is selected. Suppose we select $Op3$. Then, the end product A is determined and the process ends.

In many situations more than one operation is executable, e.g. in the first step of the example we could have chosen for $Op2$ in stead of $Op1$, which would have led to the end product immediately, but also could have had different outcomes in terms of performance (e.g. cost, throughput time). For example, when the processing time of $Op2$ is less than the processing time of $Op1$, selecting $Op2$ as a next step would perhaps be a better decision. Now, the question arises how to select the best operation from the set of executable operations to proceed. We define ‘best’ with respect to the single case, i.e. the performance goal of the case in isolation (e.g. cost, total processing time) is optimized.

2.2 Execution Strategies

We have identified several selection strategies to find the best candidate from the set of enabled operations. These strategies are related to the attributes and

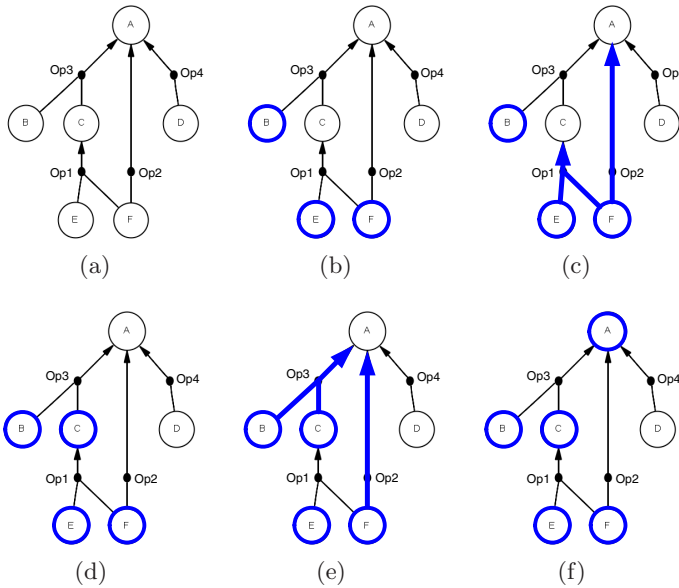


Fig. 1. The execution of a PDM: (a) The PDM, (b) Input data elements (B, E, F) are available, (c) Executable operations in the first step, (d) Data element C is produced, (e) Executable operations in step two, (f) The end product (A) is determined

properties of the operations. We drew inspiration from *sequencing and scheduling rules* in the field of logistics and production planning [3,6]:

- Random - The best candidate is randomly selected (cf. Random [3]).
- Lowest cost - The best candidate is the operation with the lowest cost.
- Shortest processing time - The operation with the shortest duration is chosen (cf. SPT [3]).
- Distance to root element - The distance of an operation to the root element is the ‘shortest path’ from this operation to an operation that produces the root element. The distance to the root element can be measured as the minimal number of operations to the root element (cf. FOPNR [3]).
- Shortest remaining processing time - The shortest remaining processing time is another form of the distance to the root element. In this case the processing times of the operations on the path to the root element are added up (cf. SR (shortest remaining processing time) [3]).

These execution strategies have been implemented in a prototype of a PBWS system. The system is able to load a PDM and generate case execution recommendations based on the PDM and the selected strategy. More details of this prototype are described in [7].

3 Conclusion

This paper presents Product Based Workflow Support: a dynamic approach to workflow execution on the basis of a product data model. In contrast to conventional workflow management support, there is no need for a process model that guides the execution. Therefore, a more dynamic and flexible support is possible. Based on the data elements readily available for a specific case on the one hand and a selected strategy (i.e. lowest cost, shortest processing time, etc.) on the other hand, this approach recommends the next step that should be performed for the case. *In contrast to conventional languages there is a clear separation of concerns: the product data model is based on functional requirements while the selected strategy focuses on performance* (e.g., minimize costs).

Using the strategies presented, the selection of the best candidate is only optimized locally (i.e. within the set of executable operations); the effect of the selected operation on future steps is not taken into account. Thus, such a strategy does not necessarily lead to the best overall path to the end product. To overcome this problem of local optimization, the use of the theory on Markov Decision Processes is a promising direction. With this analytical method, it is possible to completely compute the optimal strategy.

Acknowledgement

This research is supported by the Technology Foundation STW, applied science division of NWO and the technology programme of the Dutch Ministry of Economic Affairs.

References

1. van der Aalst, W.M.P.: On the Automatic Generation of Workflow Processes based on Product Structures. *Computers in Industry* 39, 97–111 (1999)
2. Orlicky, J.A.: Structuring the Bill of Materials for MRP. *Production and Inventory Management*, 19–42 (December 1972)
3. Panwalkar, S.S., Iskander, W.: A Survey of Scheduling Rules. *Operations Research* 25, 45–61 (1977)
4. Reijers, H.A.: Design and Control of Workflow Processes. LNCS, vol. 2617. Springer, Berlin (2003)
5. Reijers, H.A., Limam Mansar, S., van der Aalst, W.M.P.: Product-based Workflow Design. *Journal of Management Information systems* 20(1), 229–262 (2003)
6. Silver, E., Pyke, D.F., Peterson, R.: *Inventory Management and Production Planning and Scheduling*. John Wiley and Sons, Chichester (1998)
7. Vanderfeesten, I., Reijers, H.A., van der Aalst, W.M.P.: Product Based Workflow Support: A Recommendation Service for Dynamic Workflow Execution. BPM Center Report BPM-08-03, BPMcenter.org (2008)