Realtime Detection and Coloring of Matching Operator Nodes in Workflow Nets

Andreas Eckleder
Nero Development & Services GmbH, Karlsbad, Germany
aeckleder@nero.com
Thomas Freytag
Cooperative State University Karlsruhe, Germany
freytag@dhbw-karlsruhe.de
Jan Mendling
Humboldt-Universität zu Berlin, Germany
jan.mendling@wiwi.hu-berlin.de
Hajo A. Reijers
Eindhoven University of Technology, The Netherlands
h.a.reijers@tue.nl

Abstract

This work describes the implementation of an algorithm to identify and colorize matching split/join-operator pairs in workflow net based process models within the open source software WoPeD [1]. The concept was suggested as a powerful means to enhance the understandability of process graphs in [2]. The implemented detection and coloring method works in realtime, i. e. process designers get immediate feedback on actual or intended editing activities.

1 Introduction

The understandability of process graphs is a key requirement for successful visual process modelling results. In [2, 3] it was investigated how the understandability of workflow nets can be supported by several methods. One of them is to assign colors to matching pairs of control flow operators (splits and joins). The approach makes use of the fact that colors are recognized and associated with a specific semantics faster than other elements of visualization.

For a given pair of nodes in a workflow net, the number of independent paths leading from the one node to the other can be calculated with the max-flow/min-cut algorithm of Ford and Fulkerson [4]. This approach is able to determine all P/T and T/P handles of a given workflow net and therefore suitable to prove whether wellhandledness applies or not. In particular the techniques for finding matching operator nodes in a workflow net can also be applied to find mismatching operator nodes and thus can help to perform structural analysis, e. g. to check the existence or the violation of well-handledness.

In the following sections, the algorithm for performing the required check will be introduced along with its formal prerequisites. Afterwards, an implementation in the open source product WoPeD [1] will be sketched and demonstrated. Finally, a conclusion will be given with ideas to enhance the proposed techniques.

2 Approach

Our definition of a pair of matching nodes is a generalization of the concept of PT/TP-handles as used to define well-handledness in [5, 6].

In a well-handled WF-net PN, two nodes x and y are called *matching operator nodes* iff

- x is an AND-split and y an AND-join or x is an XOR-split or a place, and y an XOR-join or a place
- there is a pair of elementary paths C1 and C2 leading from x to y such that: $\alpha(C_1) \cap \alpha(C_2) = \{x, y\} \Rightarrow C_1 \neq C_2$.

The Ford and Fulkerson algorithm can be used to verify that there are indeed at least two elementary paths leading from a given node x to another node y. This can be done in analogy to the approach described in [5] to detect PT and TP handles. However, to detect all matching operator nodes of a given workflow net, all pairs of nodes $\{n_1, n_2\} \in (A_s \times A_j) \cup (X_s \times X_j) \cup (X_s \times S) \cup (S \times X_j) \cup (S \times S)$ where $A_{s/j}$ stands for the nodes of type AND-split/join respectively and $X_{s/j}$ stands for the nodes of type XOR-split/join respectively, must be checked. As only nodes with at least two elements in their postset can serve as a split and only nodes with at least two elements in their preset can serve as a join, we limit our selection of pairs to the combinations of $\{n_1, n_2\}$ where $|n_1 \cdot| > 1$ and $|\cdot n_2| > 1$. Whenever the max-flow / min-cut algorithm is reporting a maximum flow > 1 for any given pair of nodes, that pair is marked as a matching operator node. Once all matching operator pairs of a given workflow net are detected, their graphical representation can be colorized in a suitable way in order to stress the semantical relation between them. Figure 1 shows a simple example of the coloring algorithm applied to a single AND-split/join handle.

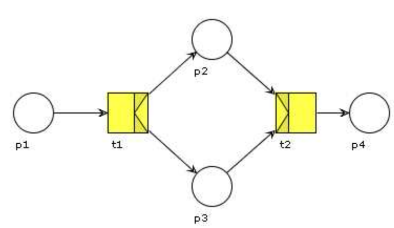

Figure 1: Simple coloring example

If multiple distinct handles exist in the same net, each matching operator node pair is assigned an individual color (see figure 2). When assigning colors to operator node pairs, it must be considered that assigning a node to a given matching operator pair is not mutually exclusive, so a given node can be part of more than one match.

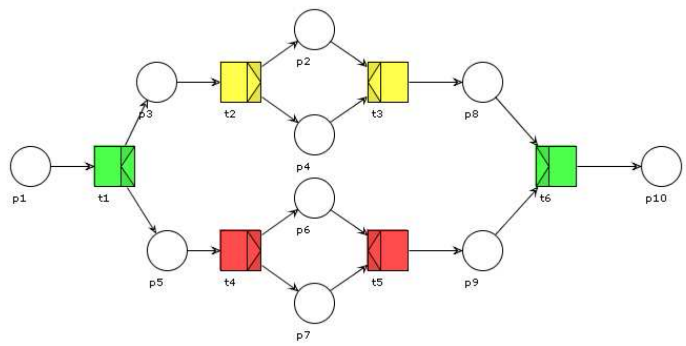

Figure 2: Example of distinct matching pairs marked in individual colors

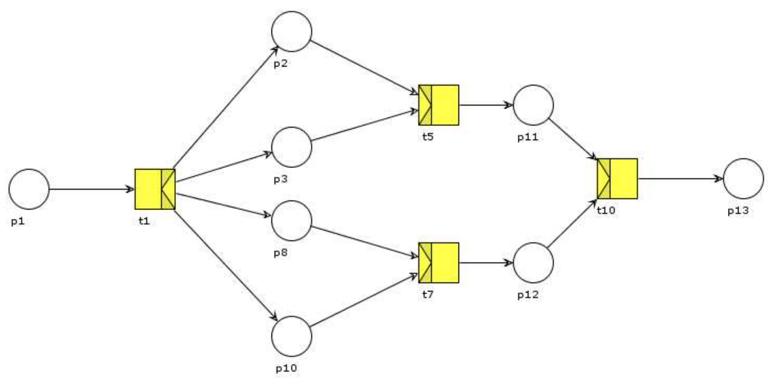

Figure 3: Handle clustering example

Since only one single color can be assigned to each node at a time, a way must be found to determine a common color for operator nodes that are part of more than one matching pair. This is done by building node clusters from the list of matching operator node pairs, where a given cluster contains all nodes of all pairs sharing at least one common node. Figure 3 shows an application example of this clustering algorithm, resulting in the same color being used for multiple matching operator handles *{t1,t7}*, *{t1,t5}* and *{t1,t10}*.

3 Implementation

WoPeD is an open source, Java-based graphical editor for workflow nets supporting the well-established "van der Aalst" notation[6]. The tool is maintained via Sourceforge, a common platform for the distributed development of free software projects. Several publications have accompanied the emerging development of WoPeD [7, 8, 9]. In the newest release which is obtainable on the

WoPeD website [1], coloring can be enabled or disabled simply by an assigned toggle button on the toolbar. When coloring is switched on, each cluster of matching operators is assigned one of the colors from a selection palette. The palette itself can be created within a settings dialog (see figure 4) and filled with arbitrary color values.

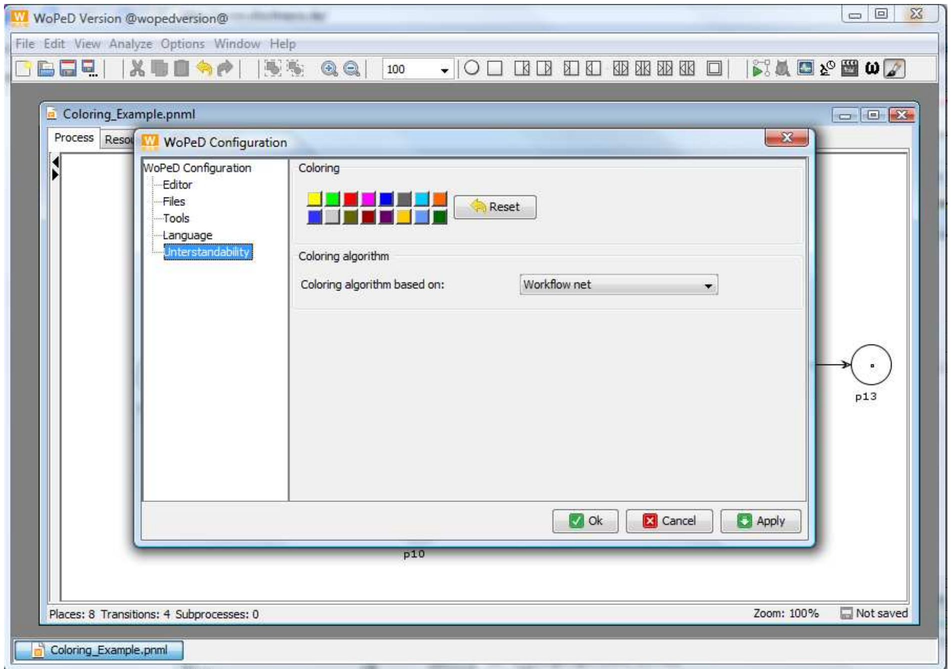

Figure 4: A settings dialog allows the configuration of optical appearance

There is a special neutral color (usually white) that is used for all nodes that have not been identified as members of any pair or cluster by the algorithm. Their graphical representation matches that of standard nodes when the coloring feature is disabled.

In "enabled" mode, the workflow net graph is constantly monitored for user-inflicted changes. If a relevant change is detected, the coloring algorithm is executed, producing a possibly new set of node clusters. Each cluster receives an individual color from the palette until all colors are in use. If this happens, colors must be re-used or the palette must be extended. Finally, the visual representation of the workflow net is updated using the new colors. To enhance the visual feedback of correct modelling, only node pairs that do not violate the rules of well-handledness are considered for coloring.

The coloring algorithm has been implemented on top of a simplified representation of the transformed workflow net. This transformed representation is derived from the original graph $G = (S, T, F)$ by inserting a first node n' and a second node n" for each node $n \in S \cup T$, and then creating an arc connecting *n'* to *n"*. Once all nodes are processed this way, an arc connecting *x"* to *y'* is inserted for all arcs $(x, y) \in F$ of the original net. The implementation of the Ford and Fulkerson algorithm is derived from the one introduced in [10], with the modification to select nodes based on "breadth-first" search. The algorithm runs at polynomial time.

Building the node clusters whose individual nodes are sharing one color has been implemented by using a simple, iterative algorithm as follows:

1. Let A be a list of sets of nodes, each set consisting of one of the matching node pairs detected
2. While $a \cap b \neq \emptyset$ for any $a, b \in A$ with $a \neq b$, set $A = A \setminus \{a, b\} + \{(a \cup b)\}$

The exemplary workflow net shown in figure 3 shows a total of three operator node pairs, each with more than one distinct node paths leading from one to another. Each pair is added to an initial list of node sets:

1. *{t1, t7}*
2. *{t1, t5}*
3. *{t1, t10}*

In the first iteration, *{t1, t7}* and *{t1, t5}* are combined to *{t1, t5, t7}*. The second and last iteration combines *{t1, t5, t7}* and *{t1, t10}* to *{t1, t5, t7, t10}*. All nodes belonging to the same set of nodes are drawn with the same color.

4 Conclusion

One shortcoming of our approach is the fact that the number of colors a human can clearly distinguish from each other is fairly limited. A possible solution for this could involve the assignment of special patterns in addition to plain palette colors (e. g. hatched, striped or plaid). Such patterns could be used to extend the amount of distinguishable handle clusters for complex workflow nets with more existing clusters than palette color entries.

The coloring algorithm has been implemented in a sufficiently generic way as to allow its application to the generalized problem of detecting PT/TP-handles and thus control-flow errors in workflow nets. Our implementation therefore also replaces the structural workflow net analysis functionality of WoPeD, allowing PT/TP-handle detection without falling back to external tools as Woflan.

References

[1] WoPeD website: www.woped.org, accessed on Aug 2, 2009.

[2] M. D. Lara. Proano: Visual layout for drawing understandable Process Models. Master's thesis, Technische Universiteit Eindhoven, 2008.

[3] J. Mendling, H.A. Reijers, and Jorge Cardoso. What Makes Process Models Understandable? In G. Alonso, P. Dadam and M. Rosemann, editors, Proceedings of the 5th International Conference Business Process Management (BPM 2007), Lecture Notes in Computer Science 4714, 48-63. Springer Verlag, Berlin, 2007.

[4] L. R. Ford and D. R. Fulkerson: Maximal flow through a network, Canad. J. Math. 8 (1956), 399-404.

[5] H. M. W. Verbeek: Verification of WF nets. PhD dissertation, Technische Universiteit Eindhoven, 2004.

[6] W. M. P. van der Aalst and K. van Hee: Workflow Management: Models, Methods, and Systems, 2002.

[7] T. Freytag and S. I. Landes: PWFtool - a Petri net workflow modelling environment. AWPN 2003 - Research Report, Catholic University of Eichstaett, 2003.

[8] C. Flender and T. Freytag: Visualizing the Soundness of Workflow Nets. AWPN 2006 - Research Report, University of Hamburg, 2006.

[9] A. Eckleder and T. Freytag: WoPeD 2.0 goes BPEL 2.0. AWPN 2008 - Research Report, University of Rostock, 2008.

[10] T. Ihringer: Diskrete Mathematik: Eine Einführung in Theorie und Anwendungen, 2002.