Ninth Int. Workshop on GNs, Sofia, 4 July 2008, 89-93

Generalized net representation of workflow nets

Ivaylo Ivanov[1], Boyan Kolev[2]

[1] – SoftConsultGroup Ltd – Dedeagach Str, Bl. 60A, c-x Strelbishte, Sofia-1408, Bulgaria
E-mail: ivaylo.ivanov@softconsultgroup.com

[2] – CLBME – Bulgarian Academy of Sciences, Bl. 105, Sofia-1113, Bulgaria
E-mail: bik@clbme.bas.bg

Abstract: The purpose of this paper is to introduce a GN representation of Workflow nets (WFGN) and all their transformation. The paper introduces important properties of WFGNs such as soundness and cyclomatic complexity.

Keywords: Workflow modeling, Generalized Nets, Workflow nets, verification, soundness, cyclomatic complexity

1. Introduction

The main purpose of workflow management systems (WFMSs) [5] is to support the definition, execution and control of workflow processes. A workflow process defines a set of activities and the specific order they are to be executed to achieve a common goal.

Business processes supported by a WFMS are centered on procedures. A procedure is the method of operation used by business process to process cases. Examples of cases are orders, claims, travel expenses, tax declarations, etc. The procedure specifies the set of tasks required to process these cases successfully. Moreover, the procedure specifies the (partial) order in which these tasks have to be executed. The goal of a procedure is to handle cases efficiently and properly. To achieve this goal, the procedure should be tuned to the ever changing environment of the business process. [1]

1.1. Generalised nets

Generalized Nets (GNs, see [3]) are defined as extensions of the ordinary Petri nets and their other modifications, but in a way that is principally different from the ways of defining the other types of Petri nets. The additional components in the GN definition give more and larger modeling possibilities and determine the place of GNs among the separate types of Petri nets, similar to the place of the Turing machine among the finite automata.

The GNs are suitable tool for describing parallel processes flowing in real-time, because these nets give the possibility to represent the process from analytical point of view (using GN-token characteristics), as well as from logical point of view (using the GN-transition condition predicates).

1.2. Workflow nets

A Place/Transition net which models the control-flow dimension of a workflow, is called a workflow net (WF-net). A WF-net can be used to specify the routing of cases. Tasks are modeled by transitions and causal dependencies are modeled by places and arcs. [2]

A workflow procedure specifies the set of tasks required to process cases successfully [1]. Workflow procedures are modeled, analyzed and verified using Petri nets. Such nets form a Petri nets subclass, called *workflow nets*. A workflow net (WF-net) is a Petri net, which satisfies the following two requirements:

(a) A WF-net has two special places – *i* and *o* – called respectively the *source* and *sink* places, which represent the enter and exit points of the workflow procedure;
(b) If we add a transition t^* to the WF-net, which connects the output place *o* with the input place *i*, the resulting Petri net is strongly connected.

A WF-net has one input place (i) and one output place (o) because any case handled by the procedure represented by the WF-net is created if it enters the WFMS and is deleted once it is completely handled by the WFMS, i.e., the WF-net specifies the lifecycle of a case. The second requirement in the definition, states that for each transition t (place p) there should be directed path from place i to o via t (p). This requirement has been added to avoid 'dangling tasks and/or conditions', i.e. tasks and conditions which do not contribute to the processing of cases [1].

The so-called soundness property of workflow as first introduced in [1] is defined as follows: For any case, the procedure will terminate eventually and the moment the procedure terminates there is a token in place o and all the other places are empty. Moreover, there should be no dead tasks, i.e., it should be possible to execute an arbitrary task by following the appropriate route though the WF-net.

In this paper we propose an approach for mapping workflow nets to generalized nets. We will give definitions of workflow generalized nets (WFGN), soundness property of WFGNs and transformation rules over WFGNs, corresponding to the ones over WF-nets. We also propose cyclomatic complexity property for WFGNs.

2. Definition of workflow generalized net (WFGN)

Definition 1: A workflow generalized net (WFGN) is a generalized net, which models a workflow procedure. A WFGN represents the corresponding WF-net and satisfies the following conditions:

(a) All predicates in an WFGN are simple (either constantly true or constantly false);
(b) A WFGN has exactly one input place *i* and one output place *o*;
(c) If we add a transition Z^* to the WFGN, connecting *o* with *i* and having only constantly true predicates in its predicate matrix, the resulting GN will have the following property:
    a. The predicates and transitions of the GN are defined in a way that for any couple of places p_1 and p_2 a token can move from p_1 to p_2 in a finite number of steps.

Definition 2 (soundness): A procedure modeled by a WFGN is sound if and only if:

(a) Place *o* is always reachable from place *i* (i.e. no infinite loops);
(b) When there is at least one token in place *o*, all other places are empty;
(c) There are no dead transitions, i.e. transitions that are never fired during the execution of the WFGN.

3. Procedure soundness verification and cyclomatic complexity

The soundness of a procedure can be verified by executing the WFGN with a number of initial states (initial tokens for the input place), monitoring the fulfillment of the three soundness conditions for each execution. The number of tests that should be made must ensure that all the cases of the procedure are considered. This number can be measured by the cyclomatic complexity [4] of the GN, which gives the maximum number of tokens that must be let into the generalized net in order to make sure that all transition predicates have been executed at least once and all places have been visited (all characteristic functions have been executed) at least once.

4. WF-net transformation rules and the corresponding GN transformations

One of the major benefits of a WFMS is the ability to change business processes very easily, i.e. without a complete redesign of the information system [1]. Four soundness preserving transformation rules for WF-nets are defined as follows [1]:

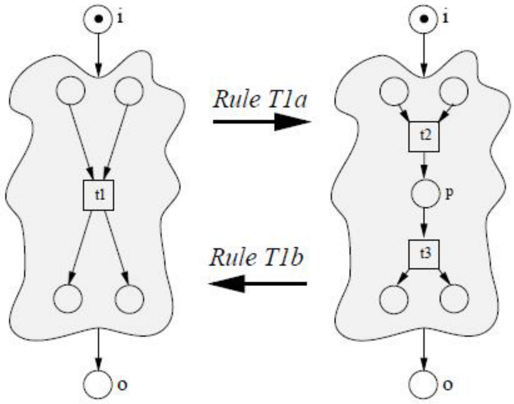

Fig.1. Transformation rules T1a and T1b

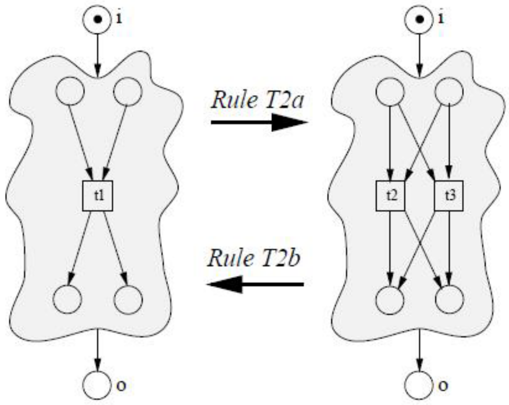

Fig.2. Transformation rules T2a and T2b

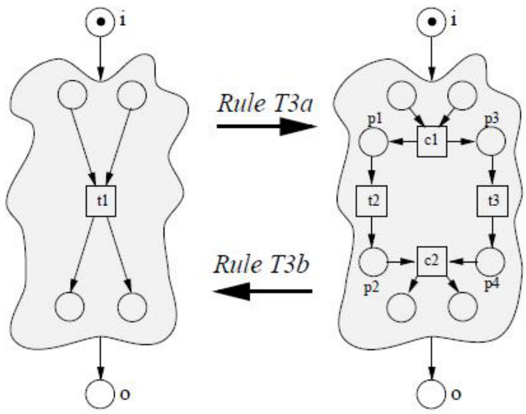

Fig.3. Transformation rules T3a and T3b

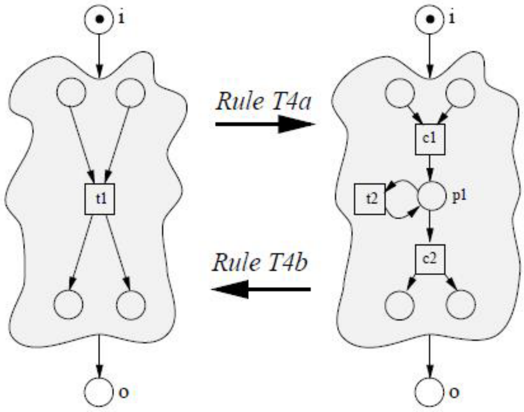

Fig.4. Transformation rules T4a and T4b

<u>T1a:</u> Task *t*1 is replaced by two consecutive tasks *t*2 and *t*3. This transformation rule corresponds to the *division* of a task: a complex task is divided into two tasks which are less complicated.
<u>T1b:</u> Two consecutive tasks *t*2 and *t*3 are replaced by one task *t*1. This transformation rule is the opposite of *T*1*a* and corresponds to the *aggregation* of tasks. Two tasks are combined into one task.
<u>T2a:</u> Task *t*1 is replaced by two conditional tasks *t*2 and *t*3. This transformation rule corresponds to the *specialization* of a task (e.g. *handle order*) into two more specialized tasks (e.g. *handle small order* and *handle large order*).
<u>T2b:</u> Two conditional tasks *t*2 and *t*3 are replaced by one task *t*1. This transformation rule is the opposite of *T*2*a* and corresponds to the *generalization* of tasks. Two rather specific tasks are replaced by one more generic task.
<u>T3a:</u> Task *t*1 is replaced by two parallel tasks *t*2 and *t*3. The effect of the execution of *t*2 and *t*3 is identical to the effect of the execution of *t*1. The transitions *c*1 and *c*2 represent control activities to fork and join two parallel threads.

T3b: The opposite of transformation rule *T3a*: two parallel tasks *t*2 and *t*3 are replaced by one task *t*1.

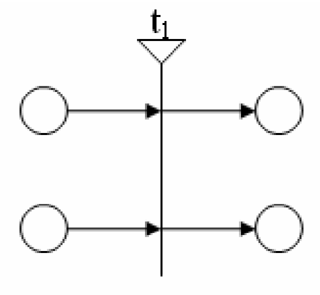

Fig.5. Initial GN construct

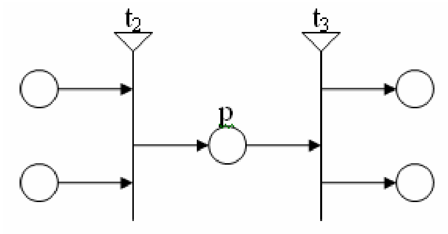

Fig.6. Transformation T1

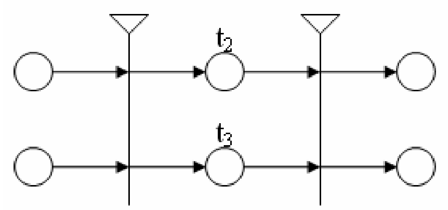

Fig.7. Transformations T2 and T3

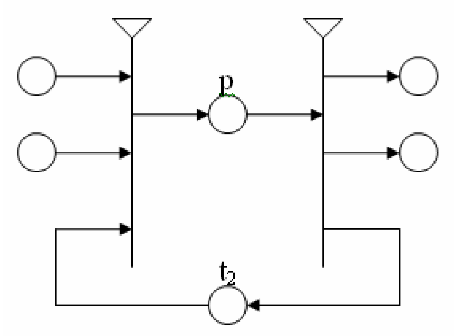

Fig.8. Transformation T4

T4a: Task *t*1 is replaced by an iteration of task *t*2. The execution of task *t*1 (e.g. *type letter*) corresponds to zero or more executions of task *t*2 (e.g. *type sentence*). The transitions *c*1 and *c*2 represent control activities that mark the begin and end of a sequence of '*t*2-tasks'. Typical examples of situations where iteration is required are quality control and communication.
T4b: The opposite of transformation rule *T4a*: the iteration of *t*2 is replaced by task *t*1.

The corresponding WFGN transformations are described below. For all of them we will use the GN depicted on Fig.5 as the initial GN (the GN construct, corresponding to the left WF-nets of the above Petri net transformations). All of the transformations are variations of the hierarchical operators H_3 and H_4 defined over generalized nets, which transform the transition t_1 to a generalized net and vice versa.

Transformation *T1* for WFGN is depicted in *Fig.6* – transition t_1 is transformed by operator H_3 to a construct of two consecutive transitions, each of which represents the two consecutive tasks t_2 and t_3.

The WFGN constructs for transformations *T2* and *T3* have identical graphical structures (*Fig.7*). The difference is in the definitions of the transition predicates and place capacities so that for transformation *T2* they allow a token to move conditionally either to t_2 or to t_3 while for transformation *T3* they allow splitting a token into both t_2 and t_3.

Transformation *T4* for WFGN is depicted in *Fig.8* – the iteration of task t_2 is performed with a loop through place *p*.

5. Conclusions

In this paper we defined workflow generalized nets (WFGN). This definition was based on mapping Workflow nets to Generalised nets. WFGNs are a robust tool for modeling business processes and workflow procedures. We presented the soundness and cyclomatic complexity properties of WFGNs that help asses the complexity of a WFGN and allow verification. We presented also transformation rules over WFGNs, corresponding to the ones over WF-nets.

References

[1] W.M.P. van der Aalst, "Verification of Workflow nets", ICATPN 97, LNCS N ° 1248, 1997.

[2] Barkaoui K., R.B. Ayed, Z. Sbai, Workflow Soundness Verification based on Structure Theory of Petri Nets, International Journal of Computing & Information Sciences, Vol. 5, No. 1, April 2007, pp. 51-61

[3] Atanassov, K., Generalized Nets, World Scientific, Singapore, 1991.

[4] Kolev, B., “Cyclomatic Complexity for Generalized Nets”, Issues in Intuitionistic Fuzzy Sets and Generalized Nets, Vol. 3 (K. Atanassov, J. Kacprzyk and M. Krawczak, Eds.), Warsaw School of Information Technology Press, 2005

[5] WFMC, "Workflow management coalition terminology and glossary (WFMC-TC-1011)", Tech. Rep., Workflow Management Coalition, Brussals, February 1999.