International Journal of Hybrid Information Technology
Vol.9, No.3 (2016), pp. 145-158
http://dx.doi.org/10.14257/ijhit.2016.9.3.14

Towards Unified Business Process Modeling and Verification for Role-based Resource-oriented Service Composition

Ming Gao[1], Lei Yang, Chunhua Zhang, Wen Guan and Ailing Li

School of Management Science & Engineering
Dongbei University of Finance & Economics, Dalian, 116025, China
gm@dufe.edu.cn

Abstract

With the prevalence of ubiquitous computing, big data, and Internet of things in cloud computing environment, it's important to consider both of collaboration, heterogeneity, isolation of multi-tenant applications and information security and privacy in service composition. Current methods need to be readdressed to cope with cross-organizational, multi-roles participated and knowledge-intensive service composition in an integrated way. Based on the modeling and verification theories of hierarchical colored petri-net, a resource-oriented collaborative workflow model, its resource control model and the joint modeling and verification method are proposed which present a unified solution bridging the gap between traditional structure-oriented workflow execution model and resource-oriented workflow domain model taking into account the underlying roles, tasks, resources and their association and coordination in design-time and runtime as well. In our approach, a business process is divided into three layers: the backbone top-level process, the task fulfillment sub-process and the task execution sub-process in order to reduce the complexity of model verification. In addition this paper gives in-depth discussions on the fine control of implicit parallel and multi-threaded process executions. Finally, the case studies show that the proposed methods are not only applicable to modeling and verification of traditional task-oriented workflows, but also suited for knowledge or data-intensive workflows which involve multi-entities complex interrelationships and relate to the domain knowledge closely.

Keywords: *Service composition, HCPN, petri-net, workflow, model verification*

1. Introduction

In recent years, Service-oriented architecture (SOA) and Service-oriented computing (SOC) have been widely adopted to build highly modular, distributed computing applications over the Internet [1]. Well-defined atomic functions encapsulated as services (e.g., Web services) can be composed into high-level business processes to conduct more complex tasks [2, 3] which is recognized as service composition [4]. Business Process Management (BPM) and workflow related theories and methodologies have gradually been investigated and adapted to service composition in-between enterprises information systems in order to standardize and ease structural processes modeling, execution control and runtime monitoring.

With the popularity of ubiquitous computing, big data, and Internet of things in modern computing environment such as hybrid clouds, it's important to consider both of collaboration, heterogeneity, isolation of multi-tenant applications and information security and privacy control of composed business processes. However, previous efforts in service composition mainly focus on separate aspects such as

[1] Corresponding author.

ISSN: 1738-9968 IJHIT

automated activities orchestration, semantic web services interoperability, workflow models' verification of correctness, etc. Current methods still need to be readdressed to cope with modeling and verification of cross-organizational, multi-roles participated and knowledge-intensive service composition in an integrated way [5].

In this paper, a Hierarchical Colored Petri-Net (HCPN) [6] based Resource-oriented Collaborative Workflow model (ROCWF), its resource control model (Task Resource Multi-role Collaboration Model, TRRC) and the joint modeling method are proposed which support tasks and resources association, allocation and coordination among multiple roles across different organizations. In order to reduce the complexity of model verification, we divide a business process into three layers including the backbone top-level process (services orchestration), the task fulfillment sub-process (constraints and relationships among roles, resources and tasks in services) and the task execution control sub-process (timer, failure recovery, *etc*.). The corresponding algorithm, principle and color set definitions used in CPN tools [7] are given as well as several case studies.

2. Related Work

There are many methods used for workflow formalization and modeling, such as: graph theory based methods (directed graph, state diagram) [8], object-oriented approach [9], formal language based method [10], State-Entity-Activity-Model (SEAM) [11], ECA (Event-Condition-Action) Rule language [12], UML (Unified Modeling Language) Modeling language [13], Information Control Net (ICN) [14], petri-net [15, 16], *etc*.

Petri-net was introduced in the field of workflow modeling and verification systematically for the first time in [17, 18]. He proposed workflow net which is an accurate formal description of workflow model and applied it to describe and validate the process of web service composition in his follow-up work.

In [19], they used HCPN to describe the process of web service composition and detect deadlock and active lock in the process, also introduced corresponding transformation rules and algorithm. Similar work can be seen in [20].

As for the industry's effort, BPMN (Business Process Management Notation) [21], a general-purpose and conceptual workflow modeling markup language is proposed as an industry standard. However, due to the lack of clear, consistent, and mathematical or graph theoretical basis, some researchers tried to use different theories and models to meet the gap. In [22], they implemented algorithms and tools to map BPMN to petri-net models.

From the short review, petri-net based theories and techniques appear to be convincing approaches to precisely model and verify structure-oriented workflow. Recent research came to pay attention to other aspects of service composition problems such as reliability, data consistency, and temporal constraints. In [23], based on petri net, they considered the transaction attributes, reliability and failure processing mechanisms to constructing reliable service composition. In [24], they added a mediation net to deal with message mismatches in service composition, furthermore generated modular timed state graphs to check the compatibility with respect to temporal constraints.

It is noteworthy that there is few research aim to put roles, tasks, resources and their interrelationships in a unified perspective to address both of structure and resource-oriented workflow modeling, as well as joint verification method.

3. Background

3.1. Petri-net

Petri-net can describe the relationships among activities such as sequential, conditional and parallel orders of execution. Some important properties including reachability, boundedness, liveness, and fairness are the key to the analysis and verification of workflow model, the same for HCPN based models.

- Reachability

$PN = (P, T, F, M_0)$ is a petri-net, if a transition $t \in T$ and a mark (state) M satisfy $M[t>M'$ ($[t>$ means t can be fired), we call M' can be reached from M directly.

If a transition list $t_1, t_2, \ldots, t_k$ and a mark list $M_1, M_2, \ldots, M_k$ satisfy $M[t_1>M_1[t_2>M_2 \ldots M_{k-1}[t_k>M_k$, we call M_k can be reached from M, or $M[\sigma>M_k$ if the transition list $t_1, t_2, \ldots, t_k$ is denoted by σ.

The set of marks which can be reached from M is noted as $R(M)$, and $M \in R(M)$.

- Home state

$PN = (P, T, F, M_0)$ is a petri-net, given $M \in R(M_0)$ (M_0 is the initial mark), if $\forall M' \in R(M)$ we can get $M \in R(M')$, we call M is a home state of PN.

- Boundedness

$PN = (P, T, F, M_0)$ is a petri-net, given a place $p \in P$, if an integer B satisfies $\forall M \in R(M_0)$: $M(p) \ll B$, we call place p is bounded, and call the minimal B as the bound of p, noted as $B(p) = min\{B \mid \forall M \in R(M_0): M(p) \ll B\}$, when $B(p)=1$, p is safe.

$PN = (P, T, F, M_0)$ is a petri-net, if every $p \in P$ is bounded, PN is a bounded petri-net, $B(PN) = max\{B(p) \mid p \in P\}$ is the bound of PN, when $B(PN) = 1$, PN is safe.

- Liveness

$PN = (P, T, F, M_0)$ is a petri-net, $t \in T$, if every $M \in R(M_0)$ and $M' \in R(M)$ satisfy $M'[t>$, the transition t is live. If every $t \in T$ is live, PN is a live petri-net.

$PN = (P, T, F, M_0)$ is a petri-net, if $M \in R(M_0)$ satisfies $\forall t \in T$: $\neg M[t>$, M is a dead mark of PN. PN is non-dead or weakly live if there is no dead mark.

- Fairness

$PN = (P, T, F, M_0)$ is a petri-net, $t_1, t_2 \in T$, if there is an integer k, $\forall M \in R(M_0)$ and $\forall \sigma \in T^*$: $M[\sigma>$ satisfy $\#(t_i / \sigma) = 0 \rightarrow \#(t_j / \sigma) \ll k$, where $i, j \in \{1, 2\}$ and $i \neq j$, we call t_1 and t_2 have a fair relationship.

If every two transitions of PN have a fair relationship, PN is a fair petri-net.

3.2. Colored Petri-net and HCPN

Colored Petri-net (CPN) [25] extends the expressiveness and transition firing mechanism of classical petri-net. It introduces color sets and multi-sets which enable the network system to describe complex data types such as list, Cartesian product, and enumeration type, as well as their multiple instances at places, transitions and guard functions which replace classical petri-net's firing rule with more powerful and customizable mechanism.

However, using CPN to model a large scale network system results in the generated single net model to be a complex graph with many elements, so that it is not easy for user to understand and maintain the model. HCPN provides an elegant and modular way to decompose a huge CPN into small subnets step by step, which allows multi-level nested and reusable subnets modeling towards a complex large scale network system [26].

4. Formalized Model Definition

4.1. Resource-Oriented Collaborative Workflow Model (ROCWF)

Definition 1. ROCWF:

$ROCWF = (P, T, F, KT, PK, G, EF, HM, EP, Ctx, M_0, M_{end}, M, E, TRRC)$, where:

N-tuple $(P, T, F, KT, PK, G, EF, M_0)$ is a CPN,

$P = \{p_1, p_2, ..., p_m\}$ is a set of places which denotes service-related or control-related states,

$T = \{t_1, t_2, ..., t_n\}$ is a set of transitions which denotes services or control actions executions,

F is the set of arcs between P (T) and T (P) in the net,

$KT = \{k_i \mid k_i \text{ is the color of } p_i, p_i \in P\}$ is non-empty color set representing all the data types involved in the net system,

$PK: P \rightarrow KT$ is a set of functions associating places with corresponding data types,

G is a list of guard functions taking control of transitions to support complex firing mechanism,

$EF: F \rightarrow KT_{ms}$ is a set of arc expression functions associating arcs with concrete color set instances flowing through,

M_0 is the initial system status.

$HM = \{SP_{top}, \{SP_{TRRC}, \{SP_{exec}\}\}\}$, where $SP \subseteq (P, T, F, KT, PK, G, EF, M_0)$ is a sub-process, defines the model's hierarchies per HCPN in which complex parts of each SP can be replaced by special transitions which act as mediations to invoke original subnets modeled in separate lower level CPNs.

$EP = PE \cup TE$ is a set of events associated with places and transitions.

Ctx is the context information perceived from the execution environment.

M_{end} is the final system status.

M is the current system status.

E is the set of execution tokens and other color set instances.

4.2. Task Resource Multi-role Collaboration Model (TRRC)

Definition 2. TRRC:

$TRRC = (R, Tasks, PT, TR, Roles, RA, TA)$, where:

$R = \{r_1, r_2, ..., r_m\}$ is a set of service-related resources.

$Tasks = \{task_1, task_2, ..., task_n\} \subseteq T$ is a set of atomic services.

$PT: P \rightarrow Tasks$ is a set of functions which associates places to the candidate tasks.

$TR: Tasks \rightarrow R = \{TR_{in}, TR_{out}, TR_{precondition}, TR_{effect}, TR_{req}, TR_{visibility}\}$ is a set of functions which defines various relationships between tasks and resources, where:

TR_{in} constrains what resources the task is allowed to read,

TR_{out} constraints what resources the task is allowed to modify,

TR_{req} constrains what resources the task must create or modify,

$TR_{precondition}$ constrains what resources must exist before the task is allowed to execute,

TR_{effect} indicates what resources may be indirectly affected after the task execution,

$TR_{visibility}$ defines the resource visibility to the task.

Obviously, for $\forall\, task_i \in Tasks$, $TR_{req}(task_i) \subseteq TR_{out}(task_i) \subseteq TR_{in}(task_i) \subseteq R$.

$Roles = \{role_1, role_2, ..., role_k\}$ represents a set of roles which can be acted as by users or service agents.

$RA: R \rightarrow Roles$ is a set of functions indicating the accessible relationship between resources and roles.

TA: Tasks→Roles is a set of functions indicating the assignment relationship between tasks and roles.

Inspired by ICN [27], the proposed TRRC model can serve as the basis for dynamic role-based resources isolation and tasks allocation in complex real-world business processes.

4.3. Hierarchical ROCWF Model

We divide a business process into three layers in order to reuse sub-processes and reduce the complexity of verification work. The CPN tools developed by University of Aarhus, Denmark is now one of the most mature HCPN modeling and simulation platform. Figure 1 shows a hierarchical workflow model implemented in CPN tools by utilizing its graphical user interface and standard ML description language.

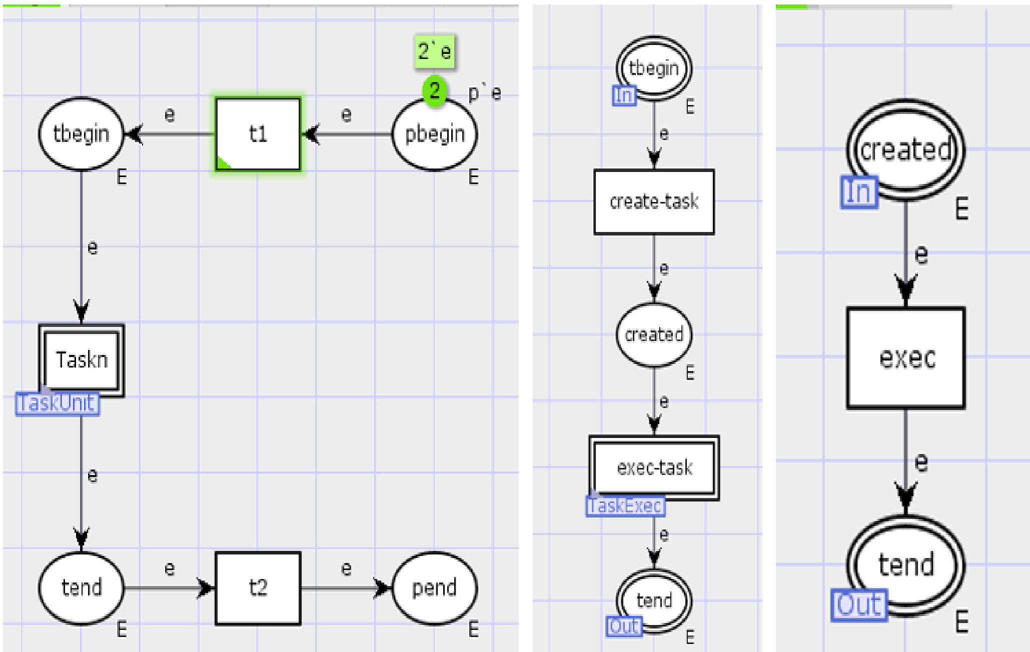


Figure 1. An Example of Three Layers of ROCWF Model (from the Left to the Right: the Backbone Top-Level Process, the Task Fulfillment sub-Process and the Task Execution Control Sub-Process)

5. Model Verification and Case Studies

In this section, based on ROCWF and CPN tools, we devise a workflow verification algorithm which identifies different layers of the model to verify corresponding properties, constraints, compatibilities, correctness, *etc*.

Algorithm 1. ROCWF Model Verification:

Step-1: For each ROCWF model, note the backbone top-level process as SP_{top}, task fulfillment sub-processes as $\{SP_{TRRC}\}$, and task execution control sub-processes as $\{SP_{exec}\}$.

Step-2: For SP_{top}, replace underlying $\{SP_{TRRC}\}$ with empty sub-processes, and introduce an empty transition (redo) to connect the beginning place and the ending place, then analyze the state space report generated by CPN tools to verify whether its reachability, boundedness, liveness, and fairness satisfy the corresponding design objectives.

Step-3: For each sub-process in *{SP_{TRRC}}*, replace underlying *{SP_{exec}}* with empty sub-processes, according to the simulation results, verify whether the sub-process's runtime behaviors satisfy the constraints in TRRC, whether the desired user, role and resources allocation and isolation are fulfilled, and whether the implicit parallel and multi-threaded tasks are under control and correctly synchronized.

Step-4: For each sub-process in *{SP_{exec}}*, verify its correctness and effectiveness on behalf of runtime security check, timeout control, failure-recovery mechanisms, etc.

Step-5: For original SP_{top}, complete simulation including all levels of sub-processes can be conducted to verify other domain specific requirements.

5.1. Verification for Backbone Top-Level Process

An example of top-level composite process with sequential, conditional and parallel execution structures is shown in Figure 2. For such a simple workflow model, it is intuitive to determine some properties such as reachability, liveness and so on with respect to HCPN by manually investigate the state space diagram generated by CPN tools as depicted in Figure 3. For instance, we can clearly see place *t2begin* and place *t3begin* both hold an execution token *e* in *state 4*, which means parallel *Task2* and *Task3* are fair. Furthermore, from each *state i*, any other *state j* can always be reached, so the process is live.

For more complex and large scale model, it is encouraged to have verification program automatically invoke the state space report function in CPN tools to completely measure reachability, boundedness, home states, liveness and fairness in detail. Figure 4 shows the results.

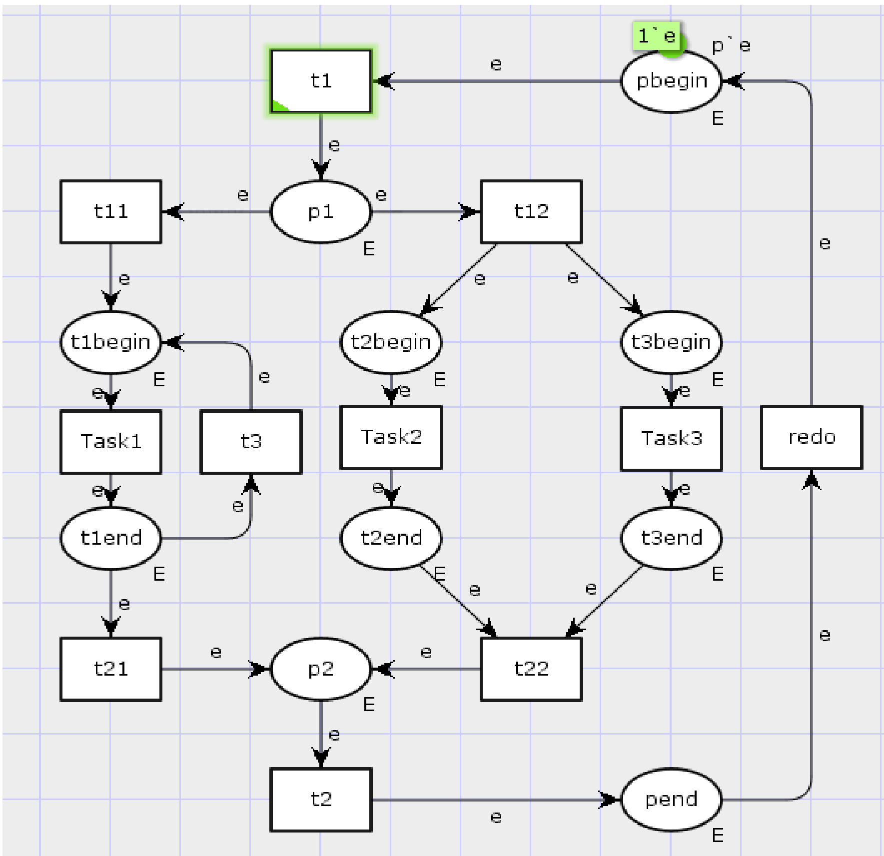


Figure 2. Backbone Top-Level Process with Complex Structures

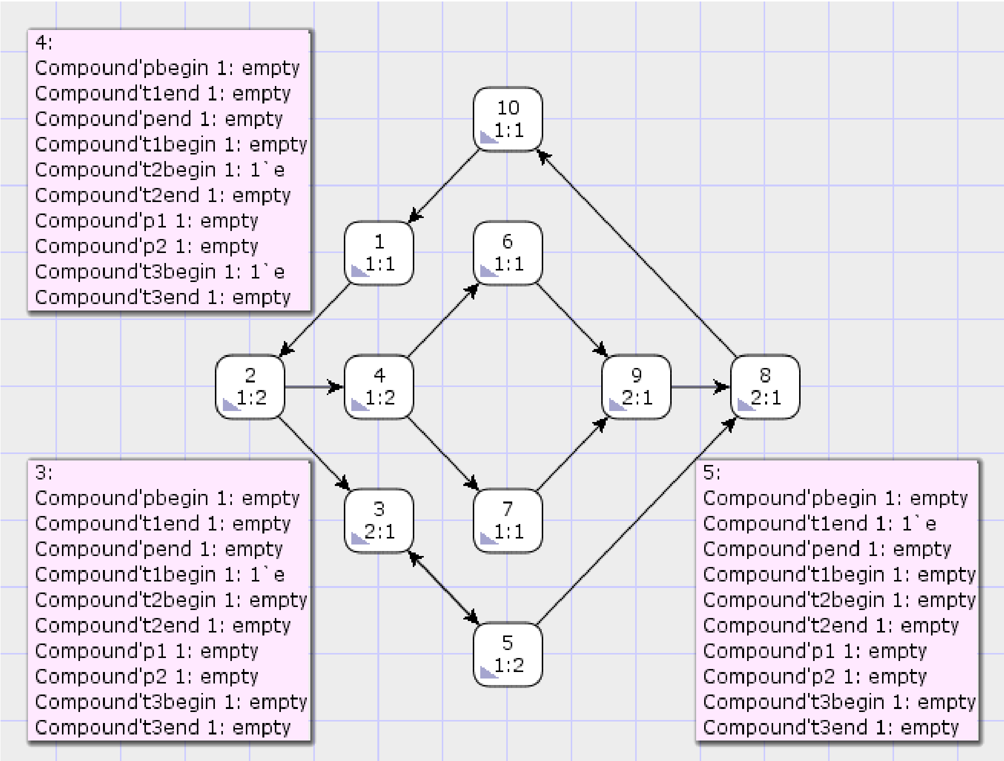


Figure 3. State Space Diagram Generated by CPN Tools for Figure 2

```
Bounded:
Best Upper Multi-set Bounds
        Compound't1begin 1  1`e
        Compound't2begin 1  1`e
        Compound't3begin 1  1`e
Home State  : All
Liveness:
Dead Markings                None
        Dead Transition Instances  None
        Live Transition Instances   All
Fairness:
Fair Transition Instances
        Compound'Task1 1
        Compound'Task2 1
        Compound'Task3 1
        Compound'redo 1
        Compound't1 1
        Compound't2 1
        Compound't22 1
…
```

Figure 4. State Space Report Generated by CPN Tools for Figure 2

5.2. Verification for Task Fulfillment Sub-Process

The verification logic of task fulfillment sub-process is illustrated in Figure 5 along with the detailed type and variable definitions in standard ML description language used in CPN tools (Table 1).

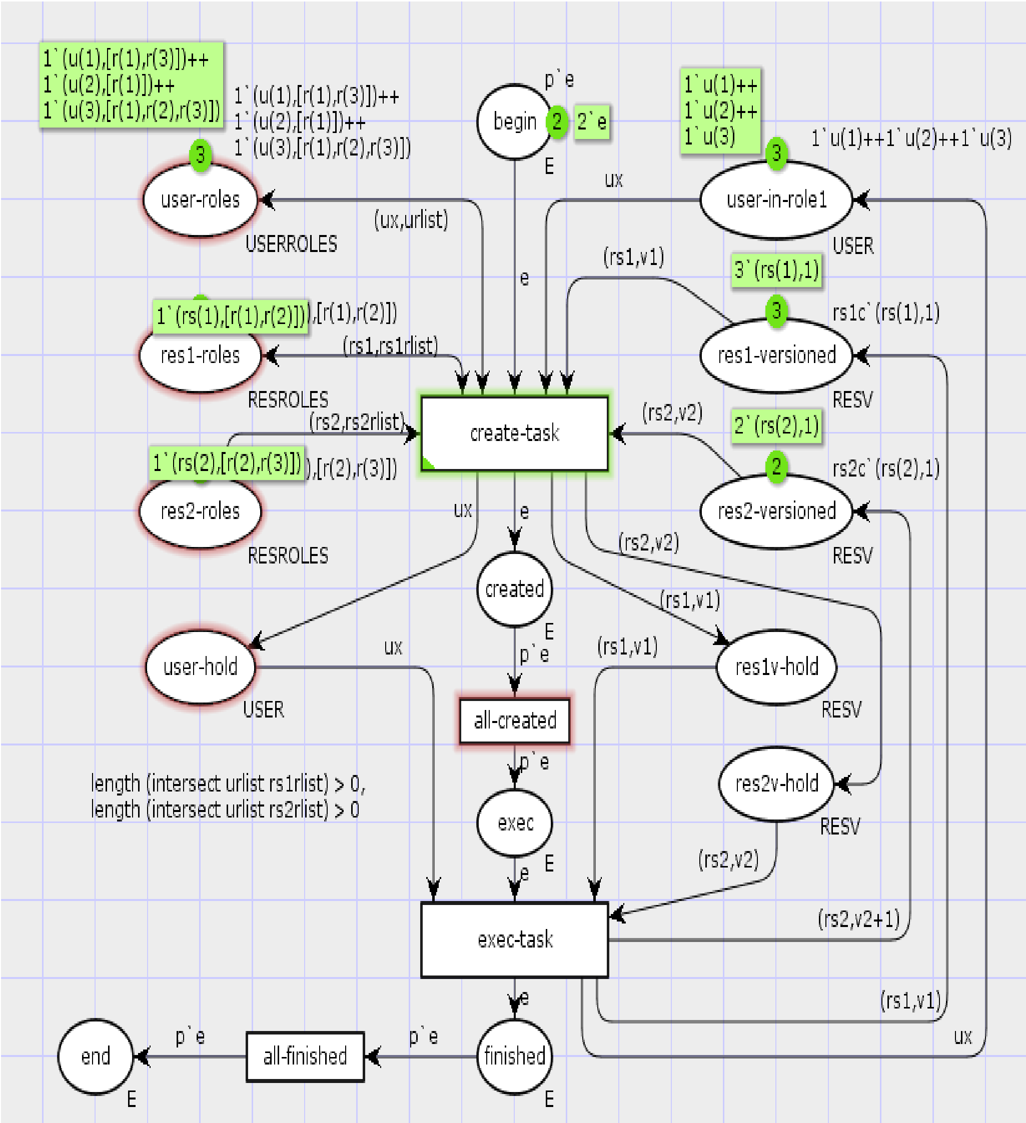


Figure 5. Task Fulfillment Sub-Process (Support TRRC Model, Implicit Parallel and Multi-Threaded Execution Verification)

The most important features are highlighted as follows:

- Dynamically assigns the task appropriate user and resources according to task-roles, task-resources, user-roles and resource-roles relationships;
- Maintains and isolates user, role and resources allocation information in task execution context; Supports resources versioning to avoid resources competition;
- Supports implicit parallel and multi-threaded tasks modeling in single sub-process, and provides different synchronization mechanisms so as to wait for all the tasks created or all the task finished with respect to specific business requirement.

Table 1. Detailed Type and Variable Definitions in CPN Tools

Type	Codes in ML

constants	val p = 2 //**number of instance excution token**
	val uc = 3 //**number of user**
	val rc = 3; //**number of role**
	val rsc = 2; //**number of resource**
	val rs1c = 3; //**number of instance allocated to resource1**
	val rs2c = 2; // **number of instance allocated to resource2**

variables	closet E = with e; //**excution token**
	closet RES = index rs with 1..rsc; //**resource index type**
	closet VERSION = int with 1..10; //**resource version**
	closet USER = index u with 1..uc; //**user index type**
	closet ROLE = index r with 1..rc; //**role index type**
	closet ROLES = list ROLE with 0..rc; //**role list type**
	closet USERROLES = product USER * ROLE; //**cartesian product of users and roles**
	closet RESROLES = product RES * ROLES; //**cartesian product of resource and roles**
	closet RESV = product RES * VERSION; //**cartesian product of resource and version**
	var ux : USER; //**user type**
	var rs1, rs2 : RES; //**resource type**
	var v1, v2 : VERSION; //**version type**
	var rs1rlist, rs2rlist : ROLES; //**role list type**
	var urlist : ROLES; //**role list type**

5.3. Verification for Task Execution Control Sub-Process

Task execution control sub-process as depicted in Figure 6 can be used to accurately simulate various scenarios such as randomized task accept or reject, task security check, task execution timeout and failure-retry or recovery to verify the correctness and effectiveness of corresponding control mechanisms.

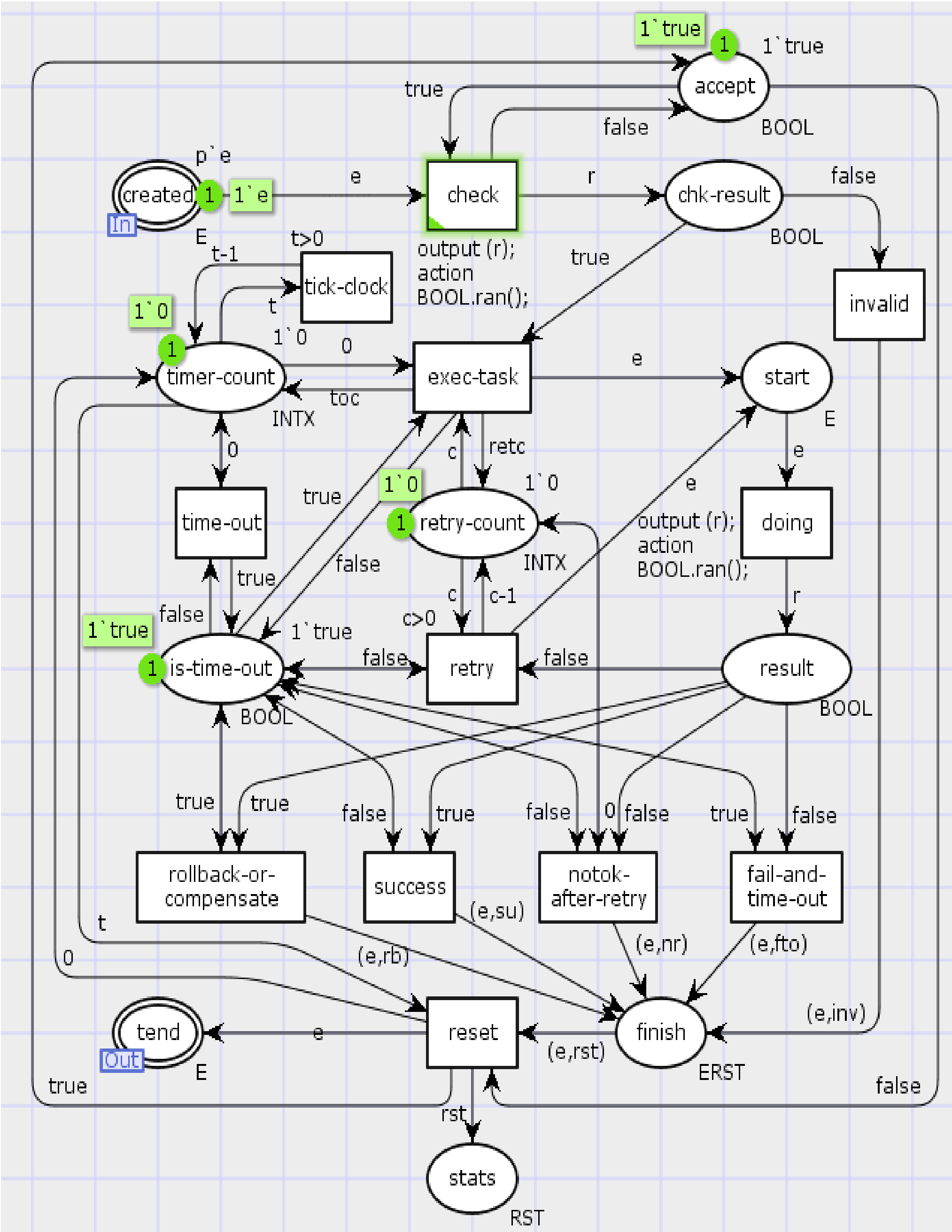


Figure 6. Task Execution Control Sub-Process

6. Conclusion and Future Work

The case studies show that the proposed methods are not only applicable to modeling and verification of traditional task-oriented workflows, but also suited for knowledge or data-intensive cross-organizational workflows which involve multi-entities complex interrelationships and relate to the domain knowledge closely.

Our ongoing work is to extend the descriptive power of proposed models to support more complex and pragmatic domain knowledge modeling considering users' needs, business goals, tasks' constraints and requirements, and devise an intelligent service planning and composition method, as well as an integrated framework unifying process planning, model verification and fine-grained execution control which uses artificial intelligence planning techniques to compose on demand business processes automatically by leveraging sequential, causal or compositional relationships of different types of entities described in the extended models to address the challenges of changeable structures and diversified users' need or business goals in real-world flexible business processes.

Acknowledgements

This work is supported by the open research project funded by Key Laboratory of Networked Control System in Chinese Academy of Sciences (2014NC001/ WLHKZ2014006), the young scholar fostering project (DUFE2014Q38) of Dongbei University of Finance & Economics (DUFE), and the science fund fostering and new technology development project from School of Management Science & Engineering (SMSE) in DUFE.

References

[1] T. Weise, M. B. Blake and S. Bleul, "Semantic Web Service Composition: The Web Service Challenge Perspective", Web Services Foundations, Springer, New York, **(2014)**, pp. 161-187.
[2] S. Dustdar and W. Schreiner, "A survey on web services composition", International journal of web and grid services, vol. 1, no. 1, **(2005)**, pp. 1-30.
[3] M. P. Papazoglou, P. Traverso, S. Dustdar and F. Leymann, "Service oriented computing: State of the art and research challenges", Computer, vol. 40, no. 11, **(2007)**, pp. 38-45.
[4] S. Song and S. W. Lee, "A goal-driven approach for adaptive service composition using planning", Mathematical and Computer Modelling, vol. 58, no. 1, **(2013)**, pp. 261-273.
[5] Q. Z. Sheng, X. Qiao, A. V. Vasilakos, C. Szabo and S. Bourne, "Web services composition: A decade's overview", Information Sciences, vol. 280, **(2014)**, pp. 218-238.
[6] K. Jensen and L. M. Kristensen, "Coloured Petri nets: modelling and validation of concurrent systems", Springer, **(2009)**.
[7] K. Jensen, L. M. Kristensen and L. Wells, "Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems", International Journal on Software Tools for Technology Transfer, vol. 9, no. 3-4, **(2007)**, pp. 213-254.
[8] Ï. B. Arpinar, U. Halici, S. Arpinar and A. Doğaç, "Formalization of workflows and correctness issues in the presence of concurrency", Distributed and Parallel Databases, vol. 7, no. 2, **(1999)**, pp. 199-248.
[9] G. Vossen and M. Weske, "The WASA2 object-oriented workflow management system", ACM SIGMOD Record, vol. 28, no. 2, **(1999)**, pp. 587-589.
[10] N. S. Glance, D. S. Pagani and R. Pareschi, "Generalized process structure grammars GPSG for flexible representations of work", Proceedings of the 1996 ACM conference on Computer supported cooperative work, **(1996)**, pp. 180-189.
[11] A. Bajaj and S. Ram, "SEAM: A state-entity-activity-model for a well-defined workflow development methodology", Knowledge and Data Engineering, IEEE Transactions on, vol. 14, no. 2, **(2002)**, pp. 415-431.
[12] J. Bae, H. Bae, S. H. Kang and Y. Kim, "Automatic control of workflow processes using ECA rules, Knowledge and Data Engineering", IEEE Transactions on, vol. 16, no. 8, **(2004)**, pp. 1010-1023.
[13] M. Dumas and A. H. M. Ter Hofstede, "UML activity diagrams as a workflow specification language", ≪UML≫2001—The Unified Modeling Language. Modeling Languages, Concepts, and Tools, Springer, Berlin Heidelberg, **(2001)**, pp. 76-90.
[14] K. Salimifard and M. Wright, "Petri net-based modelling of workflow systems", An overview, European journal of operational research, vol. 134, no. 3, **(2001)**, pp. 664-676.
[15] J. L. Peterson, "Petri Net Theory and the Modeling of Systems", Prentice-Hall, Inc., Englewood Cliffs, **(1981)**.
[16] N. R. Adam, V. Atluri and W. K. Huang, "Modeling and analysis of workflows using Petri nets", Journal of Intelligent Information Systems, vol. 10, no. 2, **(1998)**, pp. 131-158.
[17] W. M. P. Van der Aalst, "Verification of workflow nets", Application and Theory of Petri Nets, Springer, Berlin Heidelberg **(1997)**, pp. 407-426.
[18] W. M. P. Van der Aalst, "The application of Petri nets to workflow management", Journal of circuits, systems, and computers, vol. 8, no. 1, **(1998)**, pp. 21-66.
[19] Y. P. Yang, Q. P. Tan and Y. Xiao, "Verifying web services composition based on hierarchical colored petri nets", Proceedings of the first international workshop on Interoperability of heterogeneous information systems, **(2005)**, pp. 47-54.
[20] Y. W. M. Maung and A. A. Hein, "Colored Petri-Nets (CPN) based Model for Web Services Composition", IJCCER, vol. 2, no. 5, **(2005)**, pp. 169-172.
[21] S. A. White, "Introduction to BPMN", IBM Corporation, **(2004)**.

[22] R. M. Dijkman, M. Dumas and C. Ouyang, "Semantics and analysis of business process models in BPMN", Information and Software Technology, vol. 50, no. 12, **(2008)**, pp. 1281-1294.
[23] G. Fan, H. Yu, L. Chen and D. Liu, "Petri net based techniques for constructing reliable service composition", Journal of Systems and Software, vol. 86, no. 4, **(2013)**, pp. 1089-1106.
[24] Y. Du, W. Tan and M. C. Zhou, "Timed compatibility analysis of web service composition: A modular approach based on Petri nets, Automation Science and Engineering", IEEE Transactions on, vol. 11, no. 2, **(2014)**, pp. 594-606.
[25] K. Jensen, "Coloured Petri nets and the invariant-method", Theoretical computer science, vol. 14, no. 3, **(1981)**, pp. 317-336.
[26] P. Huber, K. Jensen and R. M. Shapiro, "Hierarchies in coloured Petri nets, Advances in Petri Nets", Springer, Berlin Heidelberg, **(1991)**, pp. 313-341.
[27] C. A. Ellis, "Information control nets: A mathematical model of office information flow", Proceedings of the Conference on Simulation, Measurement and Modeling of Computer Systems, vol. 3670, **(1979)**, pp. 225-240.

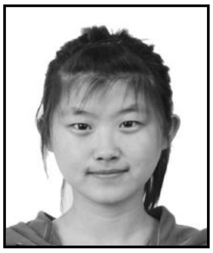
Authors

Ming Gao, He received his bachelor degree in Management Information System from Department of Economic Information in Dongbei University of Finance & Economics (DUFE) in 2002, master degree in Information Management from School of Information Engineering in DUFE in 2004, and Ph.D. in Information Technology & Management from School of Management Science & Engineering (SMSE) in DUFE in 2013. Now he is an Assistant Professor in SMSE at DUFE. His research interests include Business Process Management, Web Service Composition, Cloud Computing and Big-data Applications.

Lei Yang, He is a postgraduate student in Computer Science & Technology in the first year at School of Management Science & Engineering (SMSE) in Dongbei University of Finance and Economics (DUFE). He received his bachelor degree in Computer Science & Technology from SMSE in DUFE in 2012. His research interests include Machine Learning, Text Mining and Big-data Analysis.

Chunhua Zhang, He is an Assistant Professor in School of Management Science & Engineering (SMSE) at Dongbei University of Finance and Economics (DUFE). He is also a PhD student in Information Technology & Management at SMSE in DUFE. He received his bachelor degree from School of Management in Jilin University of Industry in 1992, and master degree from Department of Banking Management Engineering in Shanxi College of Finance & Economics in 1998. His research interests include Distributed Computing, Cloud Computing and Data Mining in Social Networks.

Wen Guan, She is an undergraduate student in Computer Science & Technology in the second year at School of Management Science & Engineering (SMSE) in Dongbei University of Finance and Economics (DUFE). Her research interests include Software Engineering and Business Process Management.



Ailing Li, She is an undergraduate student in Computer Science & Technology in the second year at School of Management Science & Engineering (SMSE) in Dongbei University of Finance and Economics (DUFE). Her research interests include Business Process Management and Data Mining.