Reset Profiles and Classical Soundness in Robustness Diagrams with Loop and Time Controls

Jasmine Malinao[1,⋆] and Richelle Ann Juayong[2]

[1] Division of Natural Sciences and Mathematics, University of the Philippines Tacloban College, Tacloban City, Leyte, Philippines,
jamalinao1@up.edu.ph,
[2] Department of Computer Science, College of Engineering, University of the Philippines, Diliman, Quezon City, Philippines

Abstract. This study establishes concepts and strategies for the verification of classical soundness of Robustness Diagrams with Loop and Time Controls(RDLT) that contain reset and join profiles. In contrast to other well-known workflow models such as BPMN, YAWL, Petri Nets, and UML Diagrams, RDLTs are powerful modelling tools that can capture all workflow dimensions, i.e. resource, process, and case, without any concerns on concept excess, the lack of support for explicit representation of data and rules, process orchestrations, and information hiding. Thus, RDLTs have found use in system representation for various real-world systems in different fields. These reset profiles in RDLTs and their mechanisms that execute system processes induce a multi-level management of information and control in its system representation. With this, previous literature had only focused on RDLTs with no resets so to avoid being encumbered by the multi-level aspect of RDLTs while performing model verification. Through this research, we provide extensions to such literature so that we are now able to manage this multi-level aspect RDLTs while also establishing their impact and relationship with various types of join and neighborhood structures in the context of verifying classical soundness of RDLTs. As a point of validation of our strategies, we benchmark the derivable activities in the input RDLT with its corresponding model abstractions. Lastly, we demonstrate and prove cases for which these abstractions help hasten the verification process or impose all abstractions to be used in the verification of the input RDLT. Finally, we provide the time and space complexity in determining classical soundness of RDLTs with such resets, joins, and neighborhood profiles, as well as pose open problems on related matters to the community.

Keywords: Resets, RDLTs, Soundness, Workflows

⋆ Corresponding author

J. Caro et al. (eds.), *Proceedings of the Workshop on Computation: Theory and Practice (WCTP 2023)*, Atlantis Highlights in Computer Sciences 20,
https://doi.org/10.2991/978-94-6463-388-7_19

1 INTRODUCTION

Robustness Diagram with Loop and Time Controls(RDLT) are multi-dimensional workflow models that had been used to represent real-world systems such as the integrated disease surveillance and response system in the country[1], Heating, Ventilation and Air-conditioning systems[2–4], and Fujitsu Ten's Computer Aided Multi-Analysis System Auto Test Tool (CATT)[11]. All the workflow dimensions, i.e. resource, process, and case, as defined in [5], can be represented in RDLT. Moreover, RDLTs also provide mechanisms to perform reset tasks in its components. This reset allows a wider capacity of control for system representation as well as volatility profiles if and when system designers would need them for system representation.

Other well-known workflow models such as the Business Process Model and Notation(BPMN)[6] and UML Diagrams, like RDLTs, do offer the capacity to represent all real-workflow dimensions in their models. However, [7] has shown the BPMN suffers from concept excess such that modellers and nontechnical users erroneously use and/or interpret another notation in BPMN in place of the appropriate one, errors in the use of multi-merges and the presence of deadlocks, and challenges in information hiding, among others. Meanwhile, UML Diagrams gravely lack supporting formal methods on model transformation and model verification so that we are able to establish workflow properties in them[8].

There had been techniques that are targeted to decompose RDLTs into one- or two-dimensional workflow models such as Workflow Nets, Petri Nets, a set of UML Diagrams, and matrix representations for the purpose of model simplification and/or verification of model properties[11–13, 16–18]. Model properties such soundness, free-choiceness, conflict-freeness, etc., defined for Workflow Nets and Petri Nets[5, 9, 10], had been adopted and extended for RDLTs since RDLTs were first introduced in [11]. These properties, and verification thereof, are important in the field of system design and analysis as they become mechanisms for which computer science practitioners, and in general, decision-makers of system planning, design, implementation, and deployment determine a criterion of fitness or suitability of models to certain environments and/or use. For example, the property of classical soundness in Workflow Nets require that the model representing a system observes (a) *proper termination* of every process and case it executes, as well as (b)*liveness* to ensure that every component of the model is reachable such that no deadlocks or livelocks, are present in the system. Whenever such requirements are deemed to be limiting for system designers, they can opt to create models that can be verified for relaxed soundness, i.e. they allow termination for some processes despite having other ongoing processes that has not/cannot be completed in the model. It may also be the case they opt for the other notions of soundness, e.g. weak soundness, k-soundness, up-to-k soundness, generalized soundness, lazy soundness, or easy soundness[5] for system representation depending on their system requirements and/or preference.

Recently, literature targeting specific types of soundness, e.g. classical and relaxed soundness[5], had been also adopted and defined in RDLTs[16, 19]. However, [19] only considered RDLTs without the inclusion and consideration of the presence of reset-bound subsystems(RBS) in them. Thus, it is the goal of our research to extend the concepts of the said paper to include RDLTs with RBS.

Taking inspiration from model decomposition by way of the vertex simplification of RDLTs in [11], we propose in this research an expanded technique of such simplification on RDLTs with RBS to transform them into its "simpler" abstractions. These abstractions are RDLTs themselves with no RBS, yet they preserve the connectivity and reusability of the components(or their abstractions) from the original output. We also provide in this study a mechanism to gauge the consistency of the original input RDLT to its abstractions with respect to the information on the system activities that are present among them. Furthermore, by using these RDLT abstractions, we are then able to reuse the existing literature on model verification techniques of RDLTs with no RBS to help verify for classical soundness in the input RDLT with RBS. Lastly, we establish the structures inside RDLTs with RBS that can influence the strategies on verifying classical soundness for them, e.g. type-alikeness of CAs and their escape arcs, establishing the use of CAs and PCAs, cases when one or both of these simpler abstractions becomes sufficient for verifying the input RDLT.

1.1 Workflow Nets with Resets and Soundness

Petri Nets and Workflow Nets[5] are two of the most widely known and mature tools for system modelling and model verification, albeit targeted mostly on the process and case dimension representation. There are model properties[9] for these classical models as well as similar models, e.g. YAWL (Yet Another Workflow Language)[14], that can include cancellation region that can disable or withdraw the execution of (ongoing) processes[10] in them. However, with such models having both cancellation regions and OR-joins, classical soundness becomes undecidable[14, 15]. This complexity is mainly attributed to (a) the dependence of nonlocal information that is used to trigger cancellations in the model, and (b) the semantics of the OR-joins that is not capable of realizing such information when resolving the processes leading to those joins.

In a similar note, the work in [16], where an input RDLT with RBS can be decomposed into a Workflow Net with reset arcs, avoids producing this nonlocal information in the latter due to the inherent design of RDLTs. That is, connectedness of non-RBS components that causes resets, i.e. out-bridges, in RBS components guarantees a well-established connectivity of their counterparts in these Workflow Nets. However, there is still an existing gap that is yet to be resolved in literature with regard to the replenishment of tokens in the cancellation region. Without such replenishment, this decomposition still incur certain behaviors in these Nets which diverge from that of their input RDLT.

2 RDLT MODELLING AND VERIFICATION

In this section, we revisit the fundamentals in RDLT modelling and verification. We shall mainly focus on the extraction of activities that are represented in an RDLT model of some system as established in [3]. Thereafter, we reiterate soundness of RDLT that is consistent with that of Workflow Nets. Finally, we present various techniques and results in literature on verifying such property in RDLTs as well as structural profiles that help satisfy its requirements.

2.1 Robustness Diagram with Loop and Time Controls

Definition 1. ***(Robustness Diagram with Loop and Time Controls[3])*** *A Robustness Diagram with Loop and Time Controls(RDLT) is a graph representation R of a system that is defined as $R = (V, E, T, M)$ where*

- *V is a finite set of vertices where every vertex is either a boundary or entity object, or a controller. An object corresponds to a resource, e.g. person, file system, table, etc., while a controller corresponds to a task in a system.*
- *E is finite set of arcs such where no two objects are connected to each other. Furthermore, every arc (x, y) has following attributes:*
  - *$C : E \to \Sigma \cup \{\varepsilon\}$ where Σ is a finite non-empty set of symbols and ε is the empty string. $C(x, y) \in \Sigma$ means that $C(x, y)$ is a condition that is required to be satisfied, e.g. input/output requirement, to proceed from x to y. Meanwhile, $C(x, y) = \varepsilon$ means that there is no condition imposed by (x, y) or signifies that x is the owner object of the controller y.*
  - *$L : E \to \mathbb{N}$ is the maximum number of traversals allowed on the arc.*
- *Let T be a mapping such that $T((x, y)) = (t_1, \ldots, t_n)$ for every $(x, y) \in E$ where $n = L((x, y))$ and $t_i \in \mathbb{N}$ is the time a check or traversal is done on (x, y) by some algorithm's walk on R.*
- *$M : V \to \{0, 1\}$ indicates whether $u \in V$ is a center of a RBS. A RBS is a substructure of G_u of R that is induced by a* ***center*** *$u \in V$, i.e. if $M(u) = 1$, and the set of controllers owned by u. (x, y) is an in-bridge of G_u if x is not a vertex in G_u but y is. Conversely, (x, y) is an out-bridge of G_u if x is a vertex in G_u, but y is not. Lastly, a pair of arcs (a, b) and (c, d) are type-alike with respect to y if (a, b) and (c, d) are both in/out-bridges of y, or both are not.*

If one has a multi-source and/or multi-sink RDLTs, [3] adopts the concept of extending Petri Nets into Workflow Nets for RDLTs, i.e. **extended RDLT**. That is, to extend the RDLT, create a dummy source s' and dummy sink f' vertices in V. Thereafter, add an arc (s', v) and (f_k, f') to E, for every $v \in V$ that is an original source in V and every $f_k \in V$ that is an original sink in the RDLT, $k \in \mathbb{N}$. Then, set $L(s', v) = 1$ and $C(s', v) = \varepsilon$, while $L(f_k, f') = 1$ and $\forall k \in \mathbb{N}, C(f_k, f') \in \Sigma$, and $C(f_i, f') \neq C(f_j, f')$, $i \neq j$.

Figure 1.(a) shows a RDLT with an RBS with center x_2 and its owned controllers x_3 and x_4, and the arcs that connect them. The in-bridges of x_2 are (x_1, x_2) and (x_6, x_2). They are type-alike with respect to x_2, however, (x_3, x_2) is not type-alike to any of them. Furthermore, $(x_3, x_2), (x_2, x_3), (x_2, x_4)$ are type-alike with respect to x_2. Both (x_4, x_5) and (x_4, x_6) are out-bridges of x_4. They are type-alike with respect to x_4. Note that the T-attributes of the arcs here are not reflected for less clutter on the image.

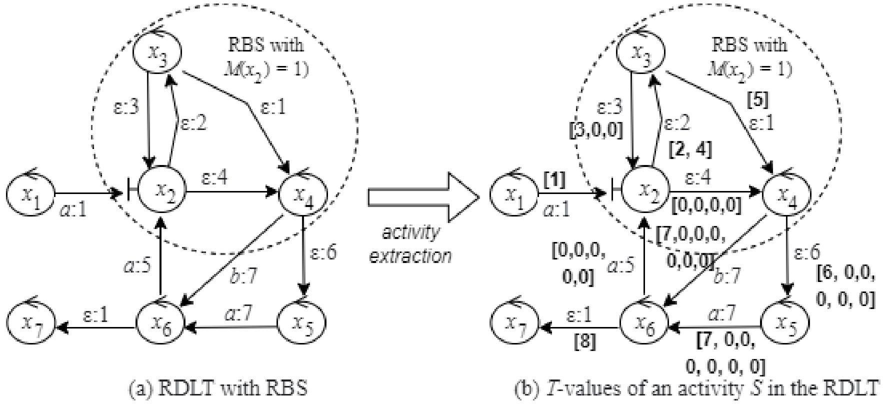


Fig. 1. RDLT with a reset-bound subsystem with center at x_3, where $M(x_2) = 1$.

The Activity Extraction Algorithm

The structure of RDLT models can capture the system components and the processes they execute. In [3], an activity extraction algorithm is introduced so as to extract activities where each activity, given its starting source vertex $s \in V$ and a target sink vertex $f \in V$, is described by its set of system components, the processes that drive it, inclusive of resets, and the specific set of cases that compose it to reach the output f. The algorithm explores the RDLT model through a depth-first search wherein a traversal from one node to its adjacent vertex is determined by satisfying conditions of their connecting arc as per its arc attribute values. This search would initially perform a *check* operation on the L- and T-values of the arc (x, y) – verifying if the number of check or traversals that had been executed on (x, y) as recorded in its T-attribute has not exceeded the maximum number of traversals, i.e. $L(x, y)$, allowed on it. If such is true, then the algorithm records the check time and proceeds to evaluate if (x, y) is **unconstrained** with respect to $(v, y) \in E$ where (x, y) and (v, y) are type-alike; otherwise, the algorithm finds another $(x, z) \in E$ to check. If (x, y) is unconstrained, then the algorithm traverses it and updates the check time with the traversal time in T; otherwise, the algorithm backtracks to x and performs another check. The check(traversal time) would always use the most recent traversal time(check time) of some (u, x) (checked (w, y)) $\in E$. If a traversal happens on an out-bridge of vertex of an RBS, then a reset of the T-attribute happens for each arc inside the RBS.

Check/Traversal times are recorded and updated in $T(x, y)$. Furthermore, traversal times, as they are final values that describe an activity derivable from R, are also recorded in $S = \{S(1), S(2), \ldots, S(n)\}$, where $S(i)$ is the set of arcs that were traversed at time i, where $(s, t) \in S(1)$, and $(q, f) \in S(n)$. Whenever resets are triggered on an RBS of R, the T-values of each arc inside the RBS are reset to their initial 0 values – as if the arc has never been checked nor traversed as per the records in T. However, these traversal records can still be retrieved via S. Hence, designers must take caution as well as take advantage of the volatile power or nature of the T-attribute of R. (This volatile power of the T-attribute is illustrated in a model representing a heating and ventilation air-conditioning system in [3] that has system components that can cycle and fluctuate their features on adsorption and desorption.)

Definition 2. ***(Unconstrained Arc[3])***
Let $R = (V, E, T, M)$ be an RDLT. An arc $(x, y) \in E$ is ***unconstrained*** *if $\forall(v, y) \in E$ where (x, y) and (v, y) are type-alike, any of the following requirements holds,*

1. *$C(v, y) \in \{\epsilon, C(x, y)\}$, i.e. the condition of (v, y) is the same as (x, y), or there are no condition imposed by (v, y) at all,*
2. *$|\{t_i \in T(x, y)|t_i \geq 1\}| \leq |\{t_j \in T(v, y)|t_j \geq 1\}| \leq L(v, y)$, i.e. the number of checks and traversals done on (x, y) has not exceeded that of (v, y)'s.*
3. *$C(v, y) \in \Sigma$, $C(x, y) = \epsilon$ and $T(v, y) \neq [0]$, i.e. there is no unsatisfied constraint $C(v, y)$. That is, (v, y) has already been checked and/or traversed by the algorithm in a prior step.*

With Definition 2, the following JOIN structures[3] can be realized for RDLTs:

1. **AND-JOIN** merging at $y \in V$: For every $(u, y), (v, y) \in E$, (v, y) and (v, y) are type-alike where $(v, y) \neq (u, y)$, their C-values $C(v, y) \neq C(u, y)$ and $C(v, y), C(u, y) \in \Sigma$.
2. **MIX-JOIN** merging at $y \in V$: There is at least one pair of type-alike arcs (u, y), $(v, y) \in E$, where $(u, y) \neq (v, y)$, $C(v, y) \neq C(u, y)$ and, without loss of generality, $C(u, y) = \epsilon$, and $C(v, y) \in \Sigma$.
3. **OR-JOIN** merging at $y \in V$: For every $(u, y), (v, y) \in E$, (u, y) and (v, y) are type-alike where $C(u, y) = C(v, y)$.

One of the activity profiles that can be extracted from the RDLT is illustrated in Figure 1.(b), i.e. $S = \{S(1), S(2), \ldots, S(8)\}$, where its reachability profiles are: $S(1) = \{(x_1, x_2)\}$, $S(2) = \{(x_2, x_3)\}$, $S(3) = \{(x_3, x_2)\}$, $S(4) = \{(x_2, x_3)\}$, $S(5) = \{(x_3, x_4)\}$, $S(6) = \{(x_4, x_5)\}$, $S(7) = \{(x_5, x_6), (x_4, x_6)\}$, $S(8) = \{(x_6, x_7)\}$. Note that (x_1, x_2) and (x_6, x_7) are unconstrained by Definition 2, requirement (1), when they were evaluated at time steps 1 and 7, respectively. Meanwhile, $(x_5, x_6)((x_4, x_6))$ is not unconstrained after its check operation at time step 6 by Definition 2, requirement (2). A backtrack to x_4 is done to ultimately reach and evaluate $(x_4, x_6)((x_5, x_6))$ to satisfy all the conditions a and b of this AND-JOIN at time step 7. For this series of traversal, the traversal of (x_4, x_5) incurs a reset of all the T-values of the arcs inside the RBS. However, the time of traversal that is stored in $S(5)$ can be retrieved to determine the next check/traversal time of the outgoing arcs of x_4.

Should the algorithm have looped back to the RBS at time step 8 by traversing (x_6, x_2), rather than terminating via (x_6, x_7), the RBS components can still be reused since their T-values were reset to 0. For example, a check (and subsequent traversal) will be successfully done on (x_2, x_3) or (x_2, x_4) at time step 9, and so on. The algorithm can reuse the RBS (components) for four(4) additional times in subsequent traversals through (x_6, x_2).

Soundness in RDLTs

In this section, we revisit soundness and verification techniques that had been formulated for RDLTs as inspired by those in literature. In particular, we focus on the existing verification of some notions of soundness for RDLTs with or without RBS. Finally, we reflect how the structures in RDLTs, e.g. its JOINs and reset-bound subsystems, impact classical or relaxed soundness for such models.

Definition 3. ***(Classical Sound*** *R)[19]*
An RDLT R is classical sound if for every activity profile $S = \{S(1), S(2), ..., S(k)\}$, $1 \leq k \leq diam(R)$, $diam(R)$ *is the diameter of R, of a set of source vertices* $I \subset V$ *and a final output vertex* $f \in V$*, where* $\forall x \in I$ *and* $y \in V, (x, y) \in S(1)$*, and* $(u, f) \in S(k), u \in V$*, the following hold,*

1. ***Proper termination***
   (a) for every $(x, y) \in S(i), i = 1, 2, \ldots, k$*,* $\exists (y, z) \in S(j)$*,* $i + 1 \leq j \leq k$*,*
   (b) for every $(x, y) \in S(k)$*,* $y = f$*.*
2. ***Liveness***
   for every $(x, y) \in E$*, there is an activity profile* $S' = \{S'(1), S'(2), \ldots, S'(k')\}$*,* $k \in N$*, such that* $(x, y) \in S'(i)$*,* $1 \leq i \leq k'$*.*

Definition 3, item (1.a) requires that every vertex y that is reached from x and included in the activity must have its subsequent child vertex z, if any, be also included in the activity and their processes eventually end up at the target output vertex f. This means that every required component of a JOIN is included in the activity if and when at least one of its component has been checked by the algorithm. That is, every pending AND- or MIX-JOIN is resolved by the algorithm upon termination. Moreover, item (1.b) requires that each vertex that is reached during the termination step is the target output vertex f of the activity profile. That is, there is no unfinished process at the termination point of the activity. Collectively, these requirements compose proper termination of R.

Lastly, item (b) requires that every edge is included in at least one activity in R. This also ensures that every component of R is usable without any incurred deadlocks (or livelocks) for any part of the model.

As for relaxed soundness of RDLTs, [19] also follows the definition of relaxed soundness for workflow nets. With reference to Definition 3, [19] weakens the first requirement, i.e. proper termination, such that R only requires an existence of at least one of its activities to observe proper termination, yet maintaining the same requirement on liveness of R.

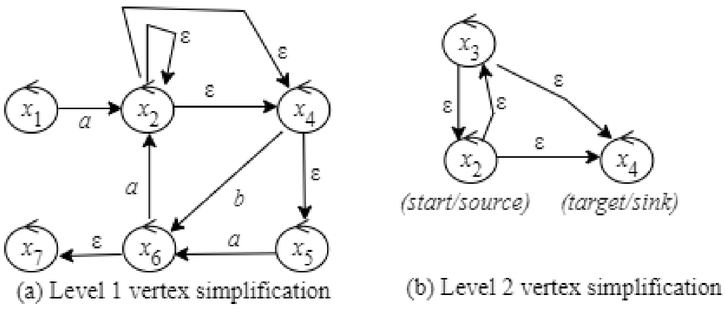
Techniques to Verify Soundness in RDLTs

Similar to some techniques of model verification, there had been several approaches that were introduced in literature to decompose, extend, and/or transform RDLTs into other models for the purpose of model simplification and/or verification, as follows,

1. **Compositional Analysis of RDLTs.**
When dealing with an RDLT R with RBS, a preprocessing step known as vertex simplification is done on R in [3] to verify soundness in R. This simplification produces two models R_1 and R_2 that are referred to as *level 1* and *level 2* vertex simplifications of R. In general, R_1 captures a subgraph of R that is found outside of its RBS as well as the vertices of the RBS that have at least 1 in-bridge or out-bridge. For each pair of vertices x and y of the RBS that are retained in R_1, an arc (x, y) is induced in R_1 if there is a path from x to y in R. Meanwhile, R_2 captures the RBS itself with vertices that have in- or out-bridges in R marked as sources and/or sinks in R_2. Whenever the activity extraction algorithm is used on R_2, these markings can be used to determine where the algorithm shall start and end, respectively. Note that a source, regardless of the C-attributes of its incoming arcs, is considered to have been already reached (through an activity extraction in R_1) at the start of the activity extraction using R_2. Whenever soundness is verified on R_1 (R_2) that contains multiple sources and/or sinks, we would always use the extended RLDT $R_1(R_2)$ as a preprocessing step for the verification process.

Note that the vertex simplification process does not modify decrease nor increase the number of loops from R to $R_1(R_2)$.

Figure 2 shows the levels 1 and 2 vertex simplifications R_1 and R_2 of R in 1, respectively. $x_2(x_4)$ is marked as the start/source(target/sink) vertex of R_2 where the activity extraction algorithm can begin(end) in its traversal process.



(a) Level 1 vertex simplification
(b) Level 2 vertex simplification

Fig. 2. The levels 1 and 2 vertex simplification R_1 and R_2 of the RDLT in Figure 1

In [3], R_1 (and R_2) (or its extended form) is used to help verify (relaxed) soundness by checking if the conditions, i.e. C-attributes of R of arcs induce deadlocks. A graph contraction strategy of $R_1(R_2)$ continually merges its vertices together by arc contraction until it results to one vertex, if possible. Two vertices x and y of the arc (x, y) of $R_1(R_2)$ are merged if the set of

C-attribute values of all arcs from x to y is a superset of the C-attributes (incl. ε) of all arcs from any other vertices in $R_1(R_2)$ to y. This means that the set of conditions and the connectivities in $R_1(R_2)$ are already satisfiable or sufficient to reach y using its set of ancestors that is represented by x through the merge process. Whenever (x, y) are merged, any incoming arc to y that is not from x has its C-attribute set to ε.

This graph contraction strategy starts at the source of $R_1(R_2)$. This strategy mimics how the C-attributes are used by the activity extraction algorithm, albeit, excluding the L- and T-attributes for a targeted model analysis.

Meanwhile, at the level of the L-attributes of R, [3] uses the L-attribute values of the arc to help verify relaxed soundness of R. The general rule for this is to ensure that each loop, whenever present in R, is disabled (i.e. not traversable) so that deadlocks in R do not happen. This is ensured by configuring the L-attribute values of the arcs in cycles to have each vertex x therein having its number of exit traversals to its child(ren) set to at most the number of entry traversals to x from its parent/s. For the case of RBS of R, this configuration of values can be generalized for the bridges and other arcs and/or paths that lead to or out of x. (For the formalism and details of this analysis of R by its L-attributes, the reader is referred to [3] regarding the property of L-verifiability of RDLTs.)

It has been shown in [19] that the combination of the two verification strategies above, i.e. C-verifiability and L-verifiability, on R is usable to prove its relaxed soundness. However, they are insufficient in proving classical soundness of R. As an example, the arcs that compose and AND-JOIN would require consistency with respect to the traversals, i.e. if one is (not) checked by the algorithm, it should also be the case that the other components are (not) checked. This matter requires restrictions on connectivity and the consistency of L-values of these arcs. Both C- and L-verifiability do not have such profiles as scope of its verification for classical soundness.

2. **Verification by Model Decomposition.**
A work done by [16] proposed a partial mapping of RDLTs with RBS into its corresponding Petri Net with reset arcs. It improved the previous work of [12] to include and map the L- and M- attributes into the resulting PN. Furthermore, this improvement in [16] posed the preservation of classical or relaxed soundness of the input RDLT, given some specific JOIN profiles, into its corresponding PN as shown in Table 1. Lastly, a significant and important contribution of [16] establishes this mapping to produce a unified model of the levels 1 and 2 vertex simplification of the RDLT. That is, the mapping does not segregate the RDLT information into its "level 1" and "level 2" PN counterparts, rather, it only produces 1 PN containing the profiles of the former models.

Input RDLT	Resulting PN
Classical sound RDLT with no MIX-JOIN	Classical Sound
Classical sound RDLT with MIX-JOIN	Relaxed Sound
Relaxed sound RDLT with no MIX-JOIN	Relaxed Sound
Relaxed sound RDLT with MIX-JOIN	Relaxed Sound

Table 1. The satisfaction of soundness for the proposed mapping of [16] on RDLTs to their output PNs.

Despite the significant contributions of the proposed mapping in [16], it also poses a certain limitations in its transformation. That is, whenever reset arcs are used in the output PNs to simulate the resets incurred by a traversal of an out-bridge in the input RDLT, there is no replenishment of tokens that can be done on the input places of the transitions that represent the components of this RBS. Note that RBS components can be reused as per their initial L- and T-values, however, their corresponding input places and transitions cannot be reached/used again in the output PN using the proposed mapping. With this, activities of the input RDLT would have a limited representation with the cases that can be simulated in the output PN.

3. **L-safeness of RDLTs without resets**
In [19], some strategies that rely on the L-attribute and JOIN structures to verify classical soundness for RDLTs, albeit without the presence of reset-bound subsystems, were provided. Nevertheless, these strategies shall be considered and extended in this paper as we move to verifying classical soundness in RDLTs with or without the presence of RBS. They do not rely on the C-attribute values themselves, however, they put restrictions on the connectivity and the L-values of RDLTs without RBS so that these models satisfy the requirements of classical soundness.

This work[19] focuses on the concept of reusability of RDLT components(i.e. vertices, arcs) and the controls that limit or enable it in the model. (Note that this work only focuses on RDLTs without RBS.) It introduced the concept of a **critical arc**(CA) of a cycle in R that drives this reuse of another component referred to as a **non-critical arc**(NCA). A critical arc of a cycle c has the least value of the L-attribute among the arcs in c, otherwise, it is an NCA of c. The reuse of a NCA $(x, y) \in E$ of R is then determinable by the sum of distinct CAs of the cycles that the NCA is involved in. (x, y) is called a ***loop-safe*** NCA if its L-attribute value is greater than this sum. Formally,

Definition 4. ***(Loop-safe NCAs, safe CAs)****[19]*
Let R be a connected RDLT with no RBS, i.e. for every vertex $v \in V$, there is a path from a source vertex $s \in V$ to v.

A cycle $c = [x_1 x_2 \ldots x_n]$ is a sequence of vertices $x_i \in V$ where $(x_i, x_{i+1}) \in E$, $i = 1, 2, \ldots, n-1$, and no two vertices in c are the same except for $x_1 = x_n$.

The elements of c is denoted as $Lit(c)$*. Moreover, the set of arcs of c is* $ArcsOfCycle(c) = \{(x,y) \in E | x_i, x_{i+1} \in Lit(c) \text{ where } x = x_i \text{ and } y = x_{i+1}\}$.

Let $Cycles(x,y) = \{cycle\ c | (x,y) \in ArcsOfCycle(c) \text{ in } R\}$.

A non-critical arc $(x,y) \in E$ *of R is* ***loop-safe*** *if* $L(x,y) > RU(x,y)$*, where* $RU(x,y) = \sum_{k=1}^{|Cycles(x,y)|} (I * L(u,v))$*, for some* $(u,v) \in minL\_CA(c_k)$*, with*

$$I = \begin{cases} 1, & \text{if } k = 1 \text{ or } \bigcup_{j=1}^{k-1} minL\_CA(c_j) \cap minL\_CA(c_k) = \varnothing, \\ 0, & \text{otherwise,} \end{cases}$$

and $minL\_CA(c) \subset E$ *is the set of arcs whose L-value is the minimum among the critical arcs found in c, accounting the other cycles* c' *in R that intersect with c.*

If there is no such cycle for (x,y)*, i.e.* $Cycles(x,y) = \varnothing$*, this means that* (x,y) *is never reused in any activity in R, i.e.* $RU(x,y) = 0$*, therefore, simply set* $L(x,y) \in \mathbb{N}$*.*
Meanwhile, a CA $(q,p) \in E$ *is called* ***safe*** *CA there is at least one NCA* $(q,r) \in E$ *of R that is loop-safe.* (q,r) *is referred to as an* ***escape arc*** *of* (q,p)*.*

Escape arcs guarantee that activities can proceed whenever (p,q) is unusable when it had already been fully used in an activity. Note that having safe CAs and loop-safe NCAs in R is insufficient in achieving classical soundness. With this, this work in [19] puts an added requirement on the JOINs in R with no RBS, i.e. they have *JOIN-safe L-values*, as follows:

Definition 5. ***(JOIN-safe L-values)****[19]*
Let $R = (V, E, T, M)$ *be a RDLT with no RBS.*

A JOIN merging at $y \in V$ *of R has* ***JOIN-safe L-values*** *if for every pair* $(u,y), (v,y) \in E$*, where* $C(u,y) \neq C(v,y)$*, the following hold:*

- *(a)* ***One split origin****. There is exactly one common ancestor* $x \in V$ *for u and v. This guarantees that the activity extraction backtracks at x from* $u(v)$ *and eventually proceeds to (tries to) resolve the JOIN by evaluating* $v(u)$*.*
- *(b)* ***No unrelated process.*** *Every process that start at the split origin x will only lead to y.*
- *(c)* ***No branching from every related process.*** *Every vertex* $q \in V$ *that is involved in a process from the split origin x to y does not have any outgoing arc* $(q,s) \in E$ *such that s does not lead to y.*
- *(d)* ***No process interruptions.*** *Every vertex* $q \in V$ *that is involved in a process from the split origin x to y is not a target vertex by some* $s \in V$*, where s is not part of such process.*

(e) ***On duplicate conditions***. *AND-JOINs merging at y should not have any pair of its arcs to have the same C-attribute value(i.e. condition). Meanwhile, MIX-JOINs can have duplicate conditions, however, no two of them are different Σ conditions.*

(f) ***Equal L-values of arcs at the AND-JOIN.*** *Every pair of arcs of an AND-JOIN merging at y must have the same L-value.*

(g) ***Loop-safe components of every related process.*** *The processes that are involved in an AND- or MIX-JOIN merging at y must have each of their component arcs to be loop-safe (NCAs). Lastly, for an OR-JOIN merging at y, each of the processes that are involved in this JOIN must be composed of loop-safe NCAs and safe CAs, if any.*

Definition 6. ***(JOIN-safe R, L-safe R)[19]***
*A RDLT R is **JOIN-safe** if every JOIN in R has its set of arcs to have JOIN-safe L-values. R is **L-safe** is every NCA is loop-safe, every CA is safe, and R is JOIN-safe.*

Theorem 1. ***(L-safeness and Classical Soundness of RDLTs)[19]***
A RDLT R with no RBS and has a single source and single sink vertex is classical sound if R is L-safe. Moreover, verifying that R is L-safe/classical sound takes $O(c|E|^4)$ and $O(|E|^2)$ time and space complexity, where c is the number of cycles in the RDLT.

The proof to Theorem 1 is detailed in [19].

The RDLT (without any RBS) in Figure 3 is L-safe.

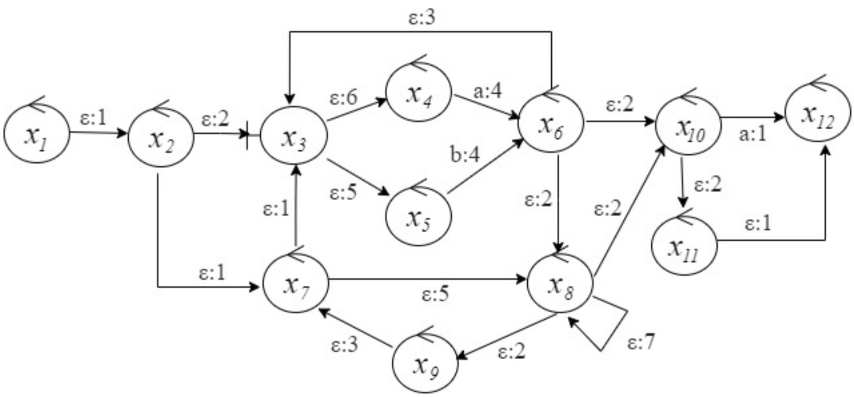


Fig. 3. A RDLT without RBS that is L-safe.(Source: [19])

In this study, we shall extend the concept of reusability of RDLT components as we now consider RDLTs with RBS. We introspect on the multi-level abstractions induced by the presence of RBS in RDLTs, the impact of type-alikeness of arcs, the reset mechanisms of RBS, and the requirements for JOIN structures to satisfy classical soundness in such RDLTs.

3 REUSABILITY AND RESETS IN RDLT

Reusability and graph connectivity, with emphasis on JOINs, of the components of RDLTs were the main drivers to verify classical soundness in RDLTs without RBS in [19]. In our study, we take some inspiration from the multi-level perspective of RDLTs whenever they have RBS in them.

For ease of discussion in the subsequent sections, we assume that R here is in its extended version, i.e. having a single, non-RBS vertex source and single-output RDLT.

3.1 Expanding the concept of Reusability

For the concept of reusability of components in R with at least one RBS, we extend the computation of $RU(x, y)$ in [19], as follows:

Definition 7. ***(Pseudocritical Arcs, Pseudo-escape Arcs)***
A ***pseudocritical arc****(PCA)* $(x, y) \in E$*, where* (x, y) *is a component of a cycle* c *in* R*, where* $L(x, y)$ *is the minimum among all the L-values of the arcs in* c *which are not arcs of an RBS in* R*.*

Meanwhile, a ***pseudo-escape arc****(PEA)* $(x, z) \in E$ *is a non-critical arc in* R *where* (x, y) *and* (x, z) *are type-alike.*

Note that a CA can also be a PCA.

We establish PCAs and their PEAs in R to provide and restore control to process flows whenever the critical arc/s of their cycles "lose" such control whenever these critical arcs are affected by resets in R. That is, when these critical arcs are found inside an RBS. In essence, the PCAs and PEAs mimic the function of critical arcs and their escape arcs as introduced in [19]. This previous literature suggested that critical arcs should not be designed within RBS to help achieve classical soundness, however, this would software engineers in building designs and their implementations. We remove this limitation through the help of PCAs and PEAs, among others, while putting additional specifications and requirements in RDLTs so that they still satisfy classical soundness. Notwithstanding, CAs remain as control points solely within the RBS as per usual.

Definition 8. ***(Expanded Reusability in RDLTs with Resets)***
Let $R = (V, E, T, M)$ *be a connected RDLT. If* $\exists v \in V$ *where* $M(v) = 1$*, then let* $B = (V', E', T', M')$ *be the RBS with its center* v*.*

Let $IBr(x)$ *be the set of in-bridges of* $x \in V$*.*

Let $Cycles_{part}(R)$ *be a set of cycles in* R *where* $\forall p = [x_1 x_2 \ldots x_n] \in Cycles_{part}(R)$*, there is at least one cycle component* $(x_i, x_{i+1}) \in E$ *that is inside an RBS*

B of R, i.e. $x_i, x_{i+1} \in V'$, and at least one cycle component $(x_j, x_{j+1}) \in E$ that is not inside B, i.e. x_j or x_{j+1} is not in V'. Whenever we have cycles $d = [d_1 d_2 \ldots d_{m_1}], e = [e_1 e_2 \ldots e_{m_2}] \in Cycles_{part}(R)$ that overlap in their non-RBS components such that this overlap contains both of their PCAs (d_k, d_{k+1}), $(e_{k'}, e_{k'+1})$, where $L(d_k, d_{k+1}) \geq (e_{k'}, e_{k'+1})$, we only consider one of these PCAs, whichever has the minimum L-value since it would ultimately contribute to the reusability of every RBS component that has a path from these cycle components. This is consistent with the computation of $RU(x,y)$ in literature. Thus, let

$$PCAs(R) = \bigcup_{c \in Cycles_{part}(R)} \{PCA\ (u,v)\ of\ c = [x_1 x_2 \ldots x_n] \in Cycles_{part}(R)|\ for\ every\ (x_i, x_{i+1}) \neq (u,v),\ where\ (x_i, x_{i+1})\ is\ a\ PCA\ of\ cycle\ c' \in Cycles_{part}(R),\ c' \neq c,\ L(u,v) \leq L(x_i, x_{i+1})\}.$$

*The **expanded reusability** $eRU(x,y)$ of $(x,y) \in E$ of R,*

$$eRU(x,y) = \begin{cases} RU(x,y), & \text{if } R \text{ has no RBS}, \\ \sum_{(u,v) \in IBr(u)} l(u,v) * (RU'(x,y) + 1), & \text{if } (x,y) \in E' \\ \text{where} & \\ l(u,v) = \min\{L(u,v), \sum_{(r,s) \in PCAs(R)\ \&\ of\ c} L(r,s)\}, & \\ \quad \text{where } (u,v), (x,y) \in ArcsOfCycle(c), & \\ \quad \text{with } c \in Cycles_{part}(R), & \\ \quad \text{or } 1 \text{ if no } c \text{ exists, and} & \\ RU'(x,y) = RU(x,y), \text{ where } RU(x,y) \text{ is} & \\ \text{computed solely wrt } E' \text{ of RBS } B, \text{and} & \\ (x,y) \text{is not a CA in } B, \text{otherwise, set} & \\ RU'(x,y) = L(x,y). & \end{cases},$$

By Definition 8, the expanded reusability $eRU(x,y)$ of the NCA arcs of R with no RBS is equal to $RU(x,y)$ as in literature. Meanwhile, if (x,y) is inside an RBS, its expanded reusability provides the maximum number of times (x,y) can be used by an activity profile in R. This $eRU(x,y)$ is determined by the number of times each cycle containing (x,y) is accessed through each of the in-bridges of its components, as well as the number of times it is (re)used inside its RBS. If and when this in-bridge is also part of this cycle, $eRU(x,y)$ uses the L-value of the PCA of this cycle since this PCA controls the (re)use of (u,v), (x,y), and the cycle itself. On another hand, if this in-bridge (u,v) is not part of any cycle that involves (x,y), yet there is a path from (u,v) to (x,y), then (x,y) is used one time in this activity through the traversal of (u,v).

3.2 Reusability in Multi-level views of RDLTs

Having expanded the concept of reusability of components in RDLTs with RBS, we can now expand the Levels 1 and 2 vertex simplification strategy to incorporate L-attribute information with considerations of the RBS behavior. Recall that the vertex simplification process leaves out the L-attribute values of the entire model. We also need to ensure that the activities that are discoverable when using the original input RDLT R (with RBS) are "preserved" in the discoverable activities when using these expanded versions of $R_1(R_2)$. For this, we first define the following criteria to determine this model preservation, as follows,

Definition 9. ***(RU-preserving transformation)***
A transformation δ of an input RDLT R to another RDLT R' is said to be ***RU-preserving*** *if for every activity profile $S = \{S(1), S(2), \ldots, S(n_1)\}$, $n_1 \in \mathbb{N}$, that is derivable from R, there is a corresponding activity profile $S' = \{S'(1), S'(2), \ldots, S'(n_2)\}$, $n_2 \in \mathbb{N}$, that is derivable from R', where for every pair $(x, y) \in S(i)$ and $(y, z) \in S(i + 1)$, either of the following holds:*

1. *$(x, y) \in S'(j)$ and $(y, z) \in S'(j + 1)$,*
2. *there exists a path $p = x_1 x_2 \ldots x_n$ in R being represented by the arc (x_1, x_n) in R' where $x_{k-1} = x, x_k, = y, x_{k+1} = z$, and*
   (a) $(s, x_1) \in S'(j)$ and $(x_1, x_n) \in S'(j + 1)$, for some vertex s in R', or
   (b) $(x_1, x_n) \in S'(j)$, for some $1 \leq j \leq n_2$, or
   (c) $(x_1, x_n) \in S'(j)$ and $(x_n, s) \in S'(j + 1)$, for some vertex s in R'.

   We call (x_1, x_n) as the ***abstract arc*** *in R' representing the path p in R.*

Since the vertex simplification of R in [19] lacks the L-values of its R_1 and R_2, we extend this process to produce the **expanded vertex simplification** R'_1 and R'_2 of R by using R_1, R_2, and R itself (to retrieve the other attribute values, e.g. L). The expanded vertex simplification R'_1 and R'_2 inherit the vertex and arc sets and attributes of R_1 and R_2, respectively, while now having either the original or derived L-values of R, as follows:

Expanded Vertex Simplification Algorithm(EVSA):
Inputs: Let $R = (V, E, T, M)$ be an connected RDLT with an RBS, i.e. $\exists v \in V$, where $M(v) = 1$. Let $R_1 = (V_1, E_1)$ and $R_2 = (V_2, E_2)$ be the vertex simplifications of R. (The C-attributes of the arcs of R, R_1, and R_2 are denoted as C, C_1, C_2, respectively, while their L-attributes are L, L_1, and L_2, respectively.)

Outputs: Level 1 and Level 2 expanded vertex simplification $R'_1 = (V'_1, E'_1, T'_1)$ and $R'_2 = (V'_2, E'_2, T'_2)$ of R, respectively . (The C-attributes of the arcs of R'_1 and R'_2 are C'_1 and C'_2, respectively, while their L-attributes are L'_1 and L'_2, respectively.)

1. For each $v \in V_1(V_2)$(or V of R), set $v' \in V'_1(V'_2)$ to be its corresponding vertex, and if $(u, v) \in E_1(E_2)$, then its there is a corresponding arc $(u', v') \in E'_1(E'_2)$, where the C-values $C'_1(u', v')(C_2(u', v'))$ of $R'_1(R'_2)$ is set to $C(u, v)$ of $R_1(R_2)$, otherwise, none.

   This step copies the vertices and arcs, inclusive of the C-values of R_1 and R_2 to their corresponding expanded versions R'_1 and R'_2, respectively.

2. For each $(u, v) \in E$, where (u, v) is not in an RBS of R, and its corresponding arc $(u', v') \in V_1'$ of L_1, let $L_1'(u', v') = L(u, v)$.

   This step copies the L-values of R for each of its arcs to the L-values of R_1' whenever such arc does not belong to an RBS of R.
3. For each abstract arc $(u', v') \in E_1'$ of R_1, where (u', v') represents the path $p = x_1 x_2 \dots x_n$ in R, i.e. $x_1 \in V$ and $x_n \in V$ correspond to $u' \in E_1'$ and $v' \in E_1'$, respectively, set $L_1'(u', v') = \min_{i=1,\dots,n-1} \{eRU(x_i, x_{i+1})\} + 1$.

   This step sets the L-value of each abstract arc (u', v') of R_1' to reflect the maximum number of reuse of the path that it represents in R. Note that we treat each component of the RBS as an NCA relative to the non-RBS components, $eRU(u', v')$ must be greater than its reusability, e.g. greater than the PCAs of the cycles (u', v') are involved in, hence, we add 1 to this maximum number of reuse. Additionally, note the eRU of the arcs along the path p can vary because of the reuse of such components within the RBS, albeit the number of times they are accessible through the in-bridges of their ancestor node is the same. However, the reusability of the entire path p itself is bound to the minimum of the L-values of the arcs therein. Thus, this minimum shall be the representative L-value for these arcs as reflected by their representative abstract arc.

For discrimination purposes, we shall refer to the L-values of abstract arcs of R_1' as **derived L-values**.

Shown in Figure 4 is the levels 1 and 2 expanded vertex simplification R_1' and R_2', respectively, of the RDLT in Figure 1. Each arc here is annotated with $C_1'(x, y) : L_1'(x, y)(eRU(x, y))$ and its arc attributes are underlined in (x, y) is a (P)CA in a cycle in R_1', R_2', and R. The PCAs of cycles $[x_2 x_4 x_5 x_6]$ and $[x_2 x_4 x_6]$ of the RDLT are the same, i.e. (x_6, x_2) (shown with underlined arc attributes in the figure). Hence, the L-value and the $eRU(x_6, x_2)$ is the same. This PCA in the original RDLT becomes a CA of its cycle in R_1'. Furthermore, every abstract arc would become NCAs in R_1' due to EVSA's computation of their derived L_1'-values, while every non-RBS components of R would retain their arc attribute values in R_1'. Lastly, the RBS of the RDLT becomes a Level 2 vertex simplification R_2' with arc attributes copied from the former to the latter. Both R_1' and R_2' become RDLTs with no RBS by themselves.

Remark 1. EVSA builds the levels 1 and 2 expanded vertex simplification R_1' and R_2' given an RDLT R in $O(|V|^2)$ time and space complexity. The bulk operation of EVSA is building the abstract arcs in R_1' by verifying if there is a path from an x to y of V, where $x(y)$ has at least one in-(out-)bridge in R.

It should be noted that each CA (or PCA) $(x, y) \in E$ should have at least one escape arc (x, z) that is type-alike with (x, y). With this, the corresponding arcs of (x, y) and (x, z) in the expanded vertex simplification will be both reflected in either the expanded vertex simplification R_1' and R_2' of R. Type-alikeness

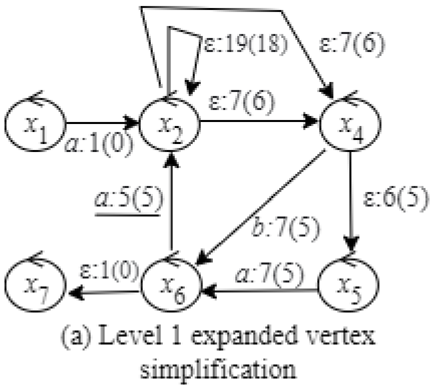


Fig. 4. The expanded vertex simplification R'_1 and R'_2 of the RDLT in Figure 1. Each arc here is annotated with $C'_1(x,y) : L'_1(x,y)(eRU(x,y))$. Furthermore, the arc attributes of (P)CAs of the cycles in R'_1, R'_2, and R are underlined for easy referencing.

ensures that both these arcs are simultaneously affected by a reset or both are not. Should they be in different levels of control, i.e. one is outside of an RBS while the other is inside, the L-attributes and T-attributes of the latter would be reverted to their initial state upon a reset as if they had never been used for traversals before, while the former do not. We prove the usefulness of type-alikenes of CAs(PCAs) and their escape arcs to achieve safe CAs (and L-safeness of R'_1) in Theorem 2 below.

Theorem 2. *Let $R = (V, E, T, M)$ be a connected RDLT with at least one RBS B. Let $(r, s) \in E$ be a critical arc of a cycle $c \in Cycles_{part}(R)$. Furthermore, let $(x, y) \in ArcsOfCycle(c)$ be the PCA of c.*

Let $R'_1 = (V'_1, E'_1, T'_1)$ and $R'_2 = (V'_2, E'_2, T'_2)$ be the levels 1 and 2 expanded vertex simplification of R.

Let $(x', y') \in E'_i$ and $(x', z') \in E'_j$, $i, j \in \{1, 2\}$, be the corresponding arcs of PCA $(x, y) \in E$ and its PEA $(x, z) \in E$, respectively, and the cycle c' of $R'_1(R'_2)$ be the corresponding arc of c.

$(x', y') \in E'_k$ is a safe CA in R'_k if there exists a loop-safe $(x', z') \in E'_k$, where $k \in \{1, 2\}$.

Proof. We first prove that (x', y') is a CA for cycle $c' = [x_1 x_2 \ldots x_n]$ in R'_k, for $k \in \{1, 2\}$. That is, for every $(x_i, x_{i+1}) \in ArcsOfCycle(c)$, $L(x', y') \le L(x_i, x_{i+1})$.

Case A: (x_i, x_{i+1}) of c' is an abstract arc in E'_1 of R'_1.

With EVSA applied on R to produce R_1' and R_2', we know that the cycle c' would have every abstract arc (x_i, x_{i+1}) being non-CAs in R_1'. That is, for every such (x_i, x_{i+1}), $eRU(x_i, x_j) > eRU(x', y')$ since (x, y) is a PCA of c in R, therefore the derived L-value $L_1'(x_i, x_{i+1}) > L_1'(x', y')$. This proves that every abstract arc $(x_i, x_{i+1}) \in E_1'$ is a non-critical arc of c' in R_1'. Therefore (x', y') is a CA in c'.

Case B: (x_i, x_{i+1}) of c' is not abstract arc in E_1' of R_1'.

For this case, we know that $L(x_i, x_{i+1}) \geq L(x', y')$ since $L(x, y)$ is the minimum L-value among all arcs in c where (x, y) is not within RBS B. Thus, (x', y') is a CA in c'.

Next, we prove that type-alikeness of $(x, y), (x, z) \in E$ of R is a requirement for $(x', y') \in E_k'$ to be a safe CA in R_k', for $k \in \{1, 2\}$.

Case 1: (x, y) and (x, z) are not type-alike.

For this case, EVSA results to having $(x', y') \in E_q'$ and $(x', z') \in E_p'$, $q, p \in \{1, 2\}$, where $q \neq p$. Since (x', y') is a CA of c' in $R_p'(R_q')$ and has no escape arc, then (x', y') is not a safe CA by Definition 4. Note that it also follows that $R_p'(R_q')$ is not L-safe by Definition 6.

Case 2: (x, y) and (x, z) are type-alike.

For this case, (x', y') and (x', z') are both in E_k' of R_k', $k \in \{1, 2\}$, where (x', z') is a CA of c' and (x', z') is its escape arc. With this, we proceed with proving the theorem for this case. That is, suppose that there no (x', z') that is a loop-safe escape arc of $(x', y') \in E_k'$ of R_k'. We know that (x', y') is not a safe CA by Definition 4.

■

Corollary 1. *For every cycle $c = [v_1 v_2 \ldots v_n]$ where every (v_i, v_{i+1}) is fully inside the RBS B of R in Theorem 2, its corresponding cycle $c' = [x_1 x_2 \ldots x_n]$ in R_2' has a safe CA $(x_j, x_{j+1}) \in E_2'$, with $1 \leq j < n$, if there exists a loop-safe escape arc $(x_j, s) \in E_2'$.*

Proof. Follows from Theorem 2.■

From hereon, we discuss how we further use the expanded vertex simplifications R_1' and R_2' of R as a strategy to help verify classical soundness in R. Note that R_1' and R_2' are already RDLTs without RBS. Hence, the technique to verify classical soundness for RDLTs without RBS in literature can be reused for verifying classical soundness of $R_1'(R_2')$ as well.

3.3 A View on JOINs in RDLTs with RBS

We now analyze the impact of AND-, MIX-, and OR-JOINs in RDLTs with RBS in relation to verification of classical soundness in these models. More specificically, we shall look at how these different types of JOINs are transformed into the components of the expanded vertex simplification of their models. Furthermore, we shall also establish the case where the expanded vertex simplification R'_1 is sufficient in verifying classical soundness of R. In literature, model properties, such as relaxed soundness, would use either R as a whole, and/or both the vertex simplifications R_1 and R_2.

Theorem 3. *Let $R = (V, E, T, M)$ be a connected RDLT with at least one RBS. Furthermore, let $R'_1 = (V'_1, E'_1, T'_1)$ and $R'_2 = (V'_2, E'_2, T'_2)$ be the expanded vertex simplification of R where R'_1 and R_2 have loop-safe NCAs and safe CAs (with a one source and one output vertex each.)*

EVSA is a RU-preserving transformation of R to R'_1 if R has no AND- or MIX-JOINs in each of its RBS.

Proof. We prove this by contradiction. Assume that $\exists R$ that has an AND-JOIN for which EVSA is an RU-preserving transformation of R to its expanded vertex simplification R'_1.

Note that since R'_1 and R'_2 have loop-safe NCAs and safe CAs, we know that for every check (and subsequent traversal) of $(x, y) \in E$ of R that is not impeded by its $L(x, y)$, there is a corresponding check (and subsequent traversal) of the corresponding (abstract) arc (x', y') of R'_1 (or R'_2) that is not impeded by its $L'_1(x', y')$, where (x', y') represents (x, y). If and when (x', y') is a PCA, we also know that it has at least one PEA to redirect the flow of traversals if $L'_1(x', y')$ already disallows checking. This behavior is also true for (x, y) and for some $(x, u) \in E$ of R, regardless if it is a CA or an NCA. From hereon, we can focus on the behavior of traversals for the models solely with respect to their connectivity, splits, and JOINs.

Let $P_1 = q_1 q_2 \dots q_n$ and $P_2 = r_1 r_2 \dots r_m$ be two paths (i.e. processes) that are wholly contained in and forms an AND-JOIN within the RBS of R such that, $q_1 = r_1$, $q_n = r_m$, $C(q_{n-1}, q_n), C(r_{m-1}, r_m) \in \Sigma$, $C(q_{n-1}, q_n) \neq C(r_{m-1}, r_m)$, and $n, m \in \mathbb{N}$. Furthermore, let $(q_n, v) \in E$ where $C(q_n, v) = \varepsilon$, and (q_n, v) is an out-bridge of q_n.

Since EVSA is an RU-preserving transformation for R to R'_1, then for every activity profile $S = \{S(1), S(2), \dots, S(n_1)\}$ that we extract using R, there is a corresponding activity $S' = \{S'(1), S'(2), \dots, S'(n_2)\}$ of its R'_1 where:

1. $(q_{n-1}, q_n), (r_{m-1}, r_m) \in S(i)$ and $(q_n, v) \in S(i+1)$, since the AND-JOIN formed by (q_{n-1}, q_n) and (r_{m-1}, r_m) imposes that both of these arcs be traversed at the same time in R; and
2. the abstract arcs $(q'_1, q'_n), (r'_1, r'_m) \in S'(j)$, and $(q'_n, v') \in S'(j+1)$, where q_i, r_i, and v in V corresponds to q'_i, r'_i, and v' of V'_1, respectively.

However, we find that item (2) does not hold for such R and R'_1 since $C(q'_1, q'_n) = C(r'_1, r'_m) = \varepsilon$. That is, without loss of generality, (q'_1, q'_n) is traversed at time j with (r'_1, r'_m) impeding this traversal. Thus, $(q'_1, q'_n) \in S'(j)$ and $(q'_n, v) \in S'(j+1)$ and $(r'_1, r'_m) \notin S'(j)$. Thus, we arrive at a contradiction.

The proof for R having a MIX-JOIN within its RBS is similarly done as above. That is, for $C(q_{n-1}, q_n) \in \Sigma$ and $C(r_{m-1}, r_m) = \varepsilon$, there is an activity where item $(q_{n-1}, q_n), (r_{m-1}, r_m) \in S(i)$ and $(q_n, v) \in S(i+1)$ holds. That is, whenever (r_{m-1}, r_m) is checked first before (q_{n-1}, q_n). This checking will cause a backtrack that would eventually reach and check (q_{n-1}, q_n). Thereafter, the activity extraction algorithm finds that (q_{n-1}, q_n) is unconstrained, thereby, it traverses both (q_{n-1}, q_n) and (r_{m-1}, r_m) at time i, and subsequently, (q_n, v) at $i+1$. Meanwhile, this simultaneous traversal of their corresponding abstract arcs (q'_1, q'_n) and (r'_1, r'_m) at time j is also not true for this case, and the algorithm proceeds to traverse (q'_n, v) at time $j+1$. Similarly, we arrive at a contradiction. ■

Theorem 4. *Let R be a connected RDLT with a set of RBS where no RBS contains an AND- or MIX-JOIN structure. Furthermore, let $R'_1 = (V'_1, E'_1, T'_1)$ and $R'_2 = (V'_2, E'_2, T'_2)$ be the expanded vertex simplification of R. If R'_1 is L-safe, then R is classical sound.*

Proof. By Theorem 3, we know that EVSA is RU-preserving for R and R'_1. Since R'_2 is composed of sequential and/or OR-JOIN process flows, the activity extraction algorithm will always proceed with no backtracks incurred, therefore, the components and the order of their reachability of R is consistent with that of the components and their order of reachability of R'_1 through the abstract paths or the retained arcs from R to R'_1. Furthermore, every component of the RBS that is represented by the (abstract) arcs in R'_1 is a loop-safe NCA, as well as every OR-JOIN that it can compose is JOIN-safe since R'_1 is L-safe. With this, there is no need to verify for classical soundness of R'_2.

With Theorem 1, R is classical sound when R'_1 is L-safe. (It also follows that R'_1 is classical sound.) ■

The level 1 expanded verification R'_1 shown in Figure 4 is sufficient in establishing classical soundness of the input RDLT in Figure 1 since its RBS does not contain any AND- or MIX-JOINs. Note that this R'_1 has loop-safe NCAs and safe NCAs. The (P)CA (x_6, x_2) is a safe CA since it has at least one escape arc (x_6, x_7) that is a loop-safe NCA. Furthermore, each of its remaining NCA (x, y) is also loop-safe since its $L'_1(x, y)$ is greater than their $RU(x, y)$ in R'_1. With these, R'_1 is classical sound since it is an L-safe RDLT.

We now discuss the case when we would need to use both R'_1 and R'_2 of R to prove for classical soundness of R. It is obvious from Theorems 3 and 4 that such case happens when there is a MIX- and/or AND-JOIN in R.

Theorem 5. *A connected RDLT $R = (V, E, T, M)$ with RBS $B = (V', E', T', M')$, where $V' \subset V$, $E' \subset E$, is classical sound if $R'_1 = (V'_1, E'_1, T'_1)$ and $R'_2 = (V'_2, E'_2, T'_2)$ are both L-safe.*

Proof. In order to prove this theorem, we first look at the structures of R'_1 and R'_2 which are produced from having AND- or MIX-JOINs in R, as follows:

1. **An AND-JOIN merging at y.**
Structure 1: Every pair $(u, y), (v, y) \in E(E')$, they are type-alike with respect to y.
For this structure, the corresponding arcs (u', y') and (w', y') of (u, y) and (v, y) shall be in E'_k, $k \in \{1, 2\}$, where (w', y) is an abstract arc. With this, the structure of $(u, y), (v, y) \in E(E')$ as an AND-JOIN merging at y is consistent with the structure of $(u', y'), (w', y') \in E'_k$ as an AND-JOIN in R'_k.

Moreover, due to the type-alikeness of (u, y) and (v, y), as well as R'_1 and R'_2 being L-safe, we know that the (derived) L-attribute value of (u', y') and (w', y') are equal in $R'_k, k \in \{1, 2\}$. Thus, the reuse of (u', y') allows, imposes, and synchronizes with the reuse of (w', y') in R'_k equivalent to the reuse of (u, y) and (v, y) in R. Hence for this structure, we can say that EVSA is an RU-preserving transformation from R to R'_1 and R'_2. Thus, since all of these AND-JOINs in R'_1 and R'_2 collectively represent all such AND-JOINs in R, then we can also conclude that R is L-safe with respect to such structures. With Theorem 1, since R is L-safe, the R is classical sound.

Related notes:
Structure 2: $(u, y) \in E \backslash E'$, and $(v, y) \in E'$, where $C(u, y), C(v, y) \in \Sigma$
Although technically not an AND-JOIN in the context of RDLTs, this structure is an important point of discussion. Here (u, y) and (v, y) are not type-alike where (v, y) is a part of an RBS of R, while (u, y) is not. This structure in R is transformed into a MIX-JOIN merging at y' in R'_k, $k \in \{1, 2\}$. Although the activity extraction algorithm will ignore (v, y) when evaluating (u, y) for unconstrainedness (and vice versa), the resulting MIX-JOIN in R'_k is not fully dealt with by the algorithm as the former. That is, if the algorithm first evaluates (w', y') versus (v', y'), (w', y') shall be constrained due to $C(v', y') \in \Sigma$ while $(w', y') = \varepsilon$ in R'_1. (Note that (w', y') is the abstract arc in R'_1 representing (v, y) in R.) This causes a backtrack on w'. This means that there will be some reachability configuration derivable of an activity in R'_1 which has no equivalent activity in R (and vice versa). With this case, EVSA is not RU-preserving for RDLTs having this kind of structure. In particular, we have the following differences of activity profiles:
(a) R has an activity that has a reachability configuration where (u, y) is present, while (v, y) is not; Meanwhile, R'_1 will not have a corresponding activity to this;
(b) R'_1 has an activity where (w', y') and (u', y') simultaneously appears once while R has no corresponding activity to this.

Notwithstanding, if R'_1 and R'_2 are L-safe, these differences of activities are immaterial to the classical soundness of R. That is, given that R'_1 is L-safe, the activity in R'_1 that is discussed in (b) in the list of differences ensures that every activity in R, inclusive of the activity described in (a) in the list, is allowed to happen and will have proper termination in R. Moreover, should there be an additional arc $(q, y) \in E'$ in R, i.e. type-alike to (v, y), the AND-JOIN which shall be present in R'_2 is verifiable for L-safeness (i.e.

Structure 1 verification). This proves that R is also L-safe if R'_1 and R'_2 are L-safe. Again, by Theorem 1, since R is L-safe, the R is classical sound.

2. **A MIX-JOIN merging at $y \in V'$.**
   Case 1: Every arc $(s, y) \in E(E')$, i.e. every pair of arcs $(u, y), (v, y) \in E(E')$ are type-alike with respect to y. Same case as in Structure 1, i.e. the structure and reuse of these components of the MIX-JOIN in R is preserved in R'_k, $k \in \{1, 2\}$. Thus, R' is classical sound if R'_1 and R'_2 are L-safe with respect to this structure.

   Case 2: This MIX-JOIN has the components $(u, y) \in E \backslash E'$, and $(v, y) \in E'$, where $C(u, y) \in \Sigma$ and $C(v, y) = \varepsilon$.
   This produces the same case as in Structure 2 of item (1). Thus, R' is classical sound if R'_1 and R'_2 are L-safe with respect to this structure.

   Case 3: This MIX-JOIN has the components $(u, y) \in E \backslash E'$, and $(v, y) \in E'$, where $C(u, y) = \varepsilon$ and $C(v, y) \in \Sigma$.
   For this case, these components of this MIX-JOIN merging at y in R is transformed into R'_k, $k = 1, 2$, as an arc that is unconstrained whenever the u or v of R (or their corresponding vertex u' or v' of R'_k) is reached. Note that the activity extraction algorithm ignores (v, y) when evaluating (u, y) (and vice versa) since they are not type-alike in R. This is also true for their corresponding structure in R'_k, $k = 1, 2$. That is, R'_k forms an OR-JOIN corresponding to this MIX-JOIN in R. By Theorem Theorem 4, we know that an L-safe R'_1 with OR-JOINs have its R (with no MIX- or AND-JOINs in its RBS) to be also L-safe. Therefore, R' is classical sound if R'_1 is L-safe with respect to this structure.

Accounting all the cases and structures above, we have proved that for R that has a MIX- and/or AND-JOIN, with or without an RBS, R is classical sound if both R'_1 and R'_2 are L-safe.
■

Remark 2. Classical soundness of an input RDLT R with at least one RBS can be verified in $O(c|E^4|)$ and $O((|V| + |E|)^2)$ time and space complexity, where c is the number of cycles in R. Essentially, the time and space complexity of the EVSA transformation that produces R'_1 and R'_2 from R, i.e. $O(|V|^2)$, is smaller than the time and space complexity of verifying R'_1 and R'_2, if necessary, i.e. $O(c|E^4|)$ and $O(|E|^2)$ as per Theorem 1. For precision, we generalize the space complexity of verifying classical soundness to $O((|V| + |E|)^2)$.

4 CONCLUSIONS AND FUTURE WORK

In this research, we were able to address the existing gap in literature pertaining to the verification of classical soundness of RDLTs with a presence of reset-bound subsystems. Along the process, we were able to extend the vertex simplification concept of RDLTs that segments an input RDLT R with RBS into two levels of abstraction R_1' and R_2' without losing or wrongly adding structural and behavioral aspects of/to the former. We were also able to prove that expanded vertex simplification process preserves the information of R within R_1' and R_2' in the context of activities despite the abstractions. With the multi-level nature of RDLTs with resets, we were able to generalize the concept of critical and non-critical arcs and relate ideal configurations of their neighborhood profiles, e.g. type-alikeness to their escape arcs, L-safeness across the levels, etc., that help achieve classical soundness of R. We were able to capture and prove certain profiles for which R_1' is already representative of both R_2' and R, and those which would need both R_1' and R_2' to prove classical soundness of R. Lastly, we were able to demonstrate the time and space complexity of verifying classical soundness of RDLTs with RBS.

As a recommendation for a future work on RDLTs, we recommend that researchers formalize the concept of the other notions of soundness, e.g. weak soundness, generalized soundness, up-to-k soundness, etc., in the context of RDLTs with RBS. Thereafter, researchers can pose efficient verification strategies for such properties in RDLTs. As an input to this future work, we find that there is a need to implement a parallel implementation of the activity extraction algorithm, proceed with analogous formalisms of the concepts and strategies of the case dimension relative to certain groups of activities defined over component similarities/categories in RDLTs, among others. Furthermore, we also recommend that researchers closely look into the concept of well-handledness of Workflow Nets, formalize such property in RDLTs, and check the impact or relevance of having MIX-JOIN structures in RDLTs with RBS on this property, and finally, relate these results to classical soundness of RDLTs.

References

1. J.C.L. Lopez, M.J. Bayuga, R.A. Juayong, J. Malinao, J. Caro, and M. Tee: Workflow models for integrated disease surveillance and response systems, Theory and Practice of Computation, 2021 Taylor and Francis Group, London, ISBN 978-0-367-41473-3. (2020)
2. J. Malinao, F. Judex, T. Selke, G. Zucker, H. Adorna, J. Caro, and W. Kropatsch: Robustness Diagram with Loop and Time Controls for System Modelling and Scenario Extraction with Energy System Applications, Energy Procedia, Vol. 88, pp. 537 – 543, ISSN 1876-6102. (2016)
3. J. Malinao: On building multidimensional workflow models for complex systems modelling. Dissertation, Technische Universit ät Wien. reposiTUm. https://doi.org/10.34726/hss.2017.43523 (2017)
4. A. Rezk: A Theoretical and Experimental Investigation of Silica Gel/Water Adsorption Refrigeration Systems. Dissertation, University of Birmingham (2012)
5. van der Aalst, W.M.P.: Structural Characterizations of Sound Workflow Nets, Computing Science Reports Vol. 9623, Eindhoven University of Technology (1996)
6. Weske, M: Business Process Management: Concepts, Languages, Architectures. Springer-Verlag Berlin Heidelberg. ISBN 978-3-540-73521-2 (2007)

7. Leopold, H., Mendling, J., and Gunther, O.: What we can learn from quality issues of BPMN models from industry, IEEE Software 33(4), -9. (2015)
8. Abbas, M., Rioboo, R., Ben-Yelles C.B., and Snook, C.: Formal modeling and verification of UML Activity Diagrams (UAD) with FoCaLiZe, Journal of Systems Architecture, Volume 114, March 2021, 101911, ISSN 1383-7621 (2021).
9. van der Aalst, W., van Hee, K., ter Hofstede, A., Sidorova, N., Verbeek, H.M.W., Voorhoeve M., and Wynn, M.: Soundness of Workflow Nets with Reset Arcs, ToPNoC III, LNCS 5800, pp. 50-70, DOI 10.1007/978-3-642-04856-2_3 (2009)
10. van der Aalst, W.M.P., van Hee, K.M., ter Hofstede, A.H.M., Sidorova, N., Verbeek, H.M.W., Voorhoeve, M. and Wynn, M.T.: Soundness of Workflow Nets: Classification, Decidability, and Analysis, Formal Aspects of Computing, Vol. 23, pp. 333-363, https://doi.org/10.1007/s00165-010-0161-4 (2011)
11. J. Malinao, K. Tiu, L.M. Lozano, S. Pascua, R.B. Chua, Ma. S. Magboo, and J. Caro: A Metric for User Requirements Traceability in Sequence, Class Diagrams, and Lines-of-Code via Robustness Diagrams. Workshop of Computation, Theory, and Practices, Proceedings in Information and Communications Technology, Vol. 7, Springer 2013, pp. 50-63. ISBN 978-4-431-54435-7 (2013)
12. A. Yiu, J. Garcia, J. A. Malinao, R. Juayong: On Model Decomposition of Multidimensional Workflow Diagrams. Proceedings of the Proceedings of Workshop on Computation: Theory and Practice (WCTP). (2018)
13. R. Delos Reyes, K. Agnes, J. A. Malinao, R. Juayong. Matrix Representation and Automation of Verification of Soundness of Robustness Diagram with Loop and Time Controls. Proceedings of the WCTP. (2018)
14. Verbeek, HMW, van der Aalst, WMP, and ter Hofsted, AHM: Verifying Workflows with Cancellation Regions and OR-joins: An Approach based on invariants. Technical Report WP 156, Eindhoven University of Technology, The Netherlands. (2006)
15. Wynn, MT, van der Aalst, WMP, ter Hofstede, AHM, and Edmond, D.: Verifying Workflows with Cancellation Regions and OR-joins: An Approach based on Reset Nets and Reachability Analysis. In International Conference on Business Process Management (BPM 2006), volume 4102 of Lecture Notes in Computer Science, pp. 389-394. Springer-Verlag, Berlin (2006)
16. Sulla, C.N. and Malinao, J.: Mapping of Robustness Diagram with Loop and Time Controls to Petri Net with Considerations on Soundness. In: Kabassi, K., Mylonas, P., Caro, J. (eds) Novel & Intelligent Digital Systems: Proceedings of the 3rd International Conference (NiDS 2023). NiDS 2023. Lecture Notes in Networks and Systems, vol 784. Springer, Cham. (2023)
17. Eclipse, K., and Malinao, J.: Model Decomposition of Robustness Diagram with Loop and Time Controls to Sequence Diagrams. In: Kabassi, K., Mylonas, P., Caro, J. (eds) Novel & Intelligent Digital Systems: Proceedings of the 3rd International Conference (NiDS 2023). NiDS 2023. Lecture Notes in Networks and Systems, vol 784. Springer, Cham. (2023)
18. Calvo, G., and Malinao, J.: Mapping Hierarchies and Dependencies from Robustness Diagram with Loop and Time Controls to Class Diagram. In: Kabassi, K., Mylonas, P., Caro, J. (eds) Novel & Intelligent Digital Systems: Proceedings of the 3rd International Conference (NiDS 2023). NiDS 2023. Lecture Notes in Networks and Systems, vol 783. Springer, Cham. (2023)
19. Malinao, J., and Juayong, RA: Classical Soundness in Robustness Diagram with Loop and Time Controls, Philippine Journal of Science, Vol. 152 (6B): 2327-2342, ISSN 0031 - 7683 (2023)