computers

MDPI

*Article*

# An Applying Colored Petri Net for Computerized Accounting System and Ledger Accounts Instruction

**Chanon Dechsupa [1], Wiwat Vatanawood [1,*], Worawit Poolsawasdi [2] and Arthit Thongtak [1]**

[1] Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok 10300, Thailand; chanon.d@chula.ac.th (C.D.); arthit.t@chula.ac.th (A.T.)
[2] Faculty of Science and Art, Chanthaburi Campus, Burapha University, Chanthaburi 22170, Thailand; worawit@buu.ac.th
* Correspondence: wiwat@chula.ac.th

**Abstract:** Many learners who are not familiar with the accounting terms find blended learning very complex to understand with respect to the computerized accounting system, the journal entries process, and tracing the accounting transaction flows of accounting system. A simulation-based model is a viable option to help instructors and learners make understanding the accounting system components and monitoring the accounting transactions easier. This paper proposes a colored Petri net (CPN)-based model for the instruction of an accounting system focused on the journal entries processes, accounting modules, and accounting transaction flows. The CPN-based language and the model checking tool named CPN are used to represent the accounting system components: a chart of accounts, an account mapping profile, the journal and ledgers system, and the financial report creations. We evaluated the designed CPN models by creating the simulation cases from ground truth data of the retail department store system and the mortgage loan system, using the decision-table-based testing technique. The results show that the designed CPN model and provided simulation cases help the learners to animate, verify, trace back accounting transactions and data flows, and increase the learner's understanding.

**Keywords:** model-based simulation; model-based instruction; computerized accounting system; formal model; colored petri net; ledger account



**Citation:** Dechsupa, C.; Vatanawood, W.; Poolsawasdi, W.; Thongtak, A. An Applying Colored Petri Net for Computerized Accounting System and Ledger Accounts Instruction. *Computers* **2021**, *10*, 169. https://doi.org/10.3390/computers10120169

Academic Editor: Paolo Bellavista

Received: 17 October 2021
Accepted: 8 December 2021
Published: 12 December 2021



## 1. Introduction

Graphical representation and simulation are a viable method for instruction on complex systems. In the computerized accounting system course and accounting information system course, learners are required to know the fundamentals of an accounting system about a chart of accounts, account mapping profile, accounting transaction flows, and related financial reports. The core course objectives of the accounting system course are that the learners will be able to identify the accounting modules correctly and should be able to perform the journal entries and describe the financial reports precisely. The learners try to understand the principles of accounting systems, especially in the accounting system components, the general ledger accounts, and the journal entries process. They learn these topics in lecture classes and test the journal entries by using the computerized accounting system. However, some accounting processes in the computerized accounting system are transparent. The heterogeneous financial transactions and many accounting processes are performed automatically, and it is difficult to trace the financial transactions elaborately even if the learners have computing and accounting skills. They can consolidate the process outcomes from the financial reports only.

Model-based instruction can be applied to increase the learner understanding because the model represents the accounting components in graphical representation. It can also simulate accounting system behaviors  flows and objects containing a collection of the

accounting data values and accounting events occurring in the accounting system. In computer science courses, there are many designing tools that support the model design and simultaneously checks the correctness of the design model. We observed that CPN-based modeling tools [1–4] can be applied for modeling and simulating the accounting system. A simulation mode and verification mode of the CPN-based modeling tools have been widely used in concurrent and distributed software and hardware verification. Therefore, we propose a CPN-based model for portraying the accounting system components and for simulating journal entry situation cases with the classification tree method [5]. The learners can use the provided CPN model to studying the accounting components, journal entries, and accounting transaction flows at their own pace, and they also refine the model and model configuration as well.

The remainder of this paper is organized as follows: Section 2 presents a literature review. Section 3 describes the background regarding computer-aided instruction, accounting system, and colored Petri net. Section 4 discusses the CPN model creation, and Section 5 illustrates the CPN model's validation and evaluation. Section 6 is the study's conclusion.

## 2. Literature Reviews

Petri-net-based techniques have been used in computer science to validate a concurrent system. There is no research that directly provides a Petri-net-based technique for an instruction of the accounting system. This section briefly describes the accounting software, and the researches that proposes the Petri-net-based techniques in computer science and business process domain.

The work of [6] provides the model-checking approach for verifying the correctness of accounting transactions interfaces. They verified the accounting system by automating the specifications of the transaction pool and ledger system into a CPN model. The CPN model is automated from the BPMN models, and they are arranged in a hierarchical structure. The authors manually created the sub-nets of a chart of accounts and account mapping profile. The CPN model represents the accounting transaction pool that collects the journal entry data, but the journal entry process and accounting code mapping are not described. Moreover, the process of ledger posting and financial report creation are not considered. The proposed CPN model is inappropriate for accounting system instruction because its structure is designed for the purpose of checking the information integration and accounting information interfaces.

Rosemary et al. [7] provided Petri-net-based model verification for simulating and analyzing an accounting system. The authors concentrated on a set of requirements of a complex AIS using a Petri net model. However, the Petri net model used in this work is very simple. The proposed model is appropriate for studying the simple internal control flows. This is a result of limitation of classical Petri nets because the classical Petri net does not provide the programming language for a data object declaration, data manipulation and constraints expression. The authors should design the model using CPN or a higher Petri net, because a higher Petri net provides many features for designing and analysing the model constraints [8] such as the control gateways, data flows, and data constraints.

The work of [9] illustrates the transformation of the decision table into a CPN construct called the D-Net. The authors provided a D-net model arranged in both a flat structure and hierarchical structure. The authors demonstrated the transformation techniques of the simple decision table and generalized decision table. However, the decision rules for transforming the decision table into the CPN transitions are incontrovertible, for which the number of the CPN transitions depends on the number of the decision rules, and the conditions of such a transformation rule are directly embedded into the CPN transition. This technique is difficult to revise the CPN model when the decision table changes because it produces many CPN transitions. This problem should be redesigned by transforming the decision table into a CPN function instead. The work of [10] provides the transformation rules of a decision table onto a CPN function, in which the conditional expressions are

automated from the decision table and stored in XML format. The conditions are mapped into a CPN function. This technique does not require an adjustment of the inscription on a CPN transition. The modelers merely modify the CPN function when the decision table changes. Nevertheless, the authors should apply the advantages of an enumerated color set and a production color set of CPN tools to reducing the number of If-Then-Else statements in the CPN function.

Yohannes et al. [11] implemented a simple accounting information system and tested their system with a case study. The authors demonstrated the system's infrastructure and the application software used; the proposed system records the accounting transactions in the Excel files, but the system only supports simple accounting cases. The work of [12] surveys the problems of an accounting information integration from the user perspective. The results of this survey list the ways and priorities to resolve or response to the integration problems in the accounting system design.

To evaluate the accounting system, the work of [13] illustrates the experiments and results of the user satisfaction measurement of an accounting framework. The questionnaires are used for gathering information from students who have used the accounting framework. The evaluated framework has been implemented using Delone and McLean. The variables, dictators, and hypothesis model used for measuring the qualities of the framework are detailed, and the research variables and indicators of this work are used for evaluating the learner satisfaction of our model.

In our CPN model creation process, a chart of accounts must be expressed in the CPN ML programming language [2]. The chart of accounts is implemented with a CPN variables declaration and a CPN function. We applied the transformation techniques of [9,10] to create the chart of accounts stored in an arrayList variable and created the CPN function in terms of a decision table for mapping and validating the account codes. We enhanced the decision table of [9] as a CPN function, and it supports the requirement changes on a chart of accounts and on an account mapping profile. The work of [11] is used as additional information for the learners, describing the coordination of accounting business processes, accounting information integration, and historical financial data [14].

## 3. Background

### *3.1. Computer-Aided Instruction and Accounting Information Systems*

Computer-aided instruction or computer-assisted instruction (CAI) [15] is an interactive instructional multimedia for increasing a learner's understanding and allowing the learners to proceed at their own pace. CAI is used in various disciplines and is accordingly designed with respect to the objectives and resources of a course. It record-keeps, progresses tracking, and assesses the learner performance. The CAI program comprises the representations of course contents and their extension, including drilling, practicing, and simulating tutorials and activities. Multimedia of the CAI program can be applied in studying the fundamental accounting and develops accounting skills in problem solving.

An accounting system is the system that manages the organization's financial information. This paper focuses on the computerized accounting system or the accounting information system (AIS) [16]. The cycle of the accounting system is shown in Figure 1. The accounting process starts at the journal entry module, which is an accounting treatment operated accordingly to the principle of double-entry accounting [17]. A chart of accounts [16] is a list of financial accounts that is set up for mapping the raw financial data or business transaction into the accounting transactions. A ledger contains a list of accounting transactions derived from the journal entries. A ledger posting is a transforming process the Debit and Credit amounts in the journals into the corresponding ledger account. Next, the accountants may adjust the account at the end of account period, and the financial report creation and the book closing or ledger closing are performed, respectively. Essentially, AIS provides the seven core modules as follows:

1. Point of sale module (POS module): this represents the place where the raw financial transactions takes place. The financial transactions may be posted by hand or

automated by an integrated POS module. In the integrated POS module, the account mapping profile is used to map the raw financial transactions into the accounting transactions.

2. Chart of account module or COA: A chart of accounts is manipulated at the COA module, on which the accounts in the list are classified into five series: assets, liabilities, equity, revenue, and expenses.
3. Accounts receivable module: accounts receivable handles the invoicing and manages the customer payments.
4. Accounts payable module: accounts payable handles the recording and tracking of the money owned by a business.
5. General ledger module: the general ledger module is responsible for consolidating the accounting information from the point of sale module, accounts receivable module, and accounts payable module. Financial reports are computed from the accounting transactions stored in the general ledger module, and these transactions may be used for auditing purposes.
6. Trial balance module: the trial balance module shows a listing of accounts that come along with the net balance of each account. It is useful for an internal checking because a trial balance can disclose the net balance both before adjustment and after adjustment, and it comprises the primary data for creating the balance sheet, income statement and cash flow report.
7. Balance sheet module: this details a list of the organization assets, liabilities, and equity. It also be used to create and estimate the clash flow budget.

Although there are educational software licenses of accounting system for students, the tracing of accounting transaction flows and the checking of transformations of raw financial data into the accounting transaction can be consolidated and verified by data inspection on the finalized financial reports only. It is better for the learners to be able to interact with the accounting system simulation step-by-step, and arbitrarily trace and monitor the transaction flows and the data objects in each accounting situation.

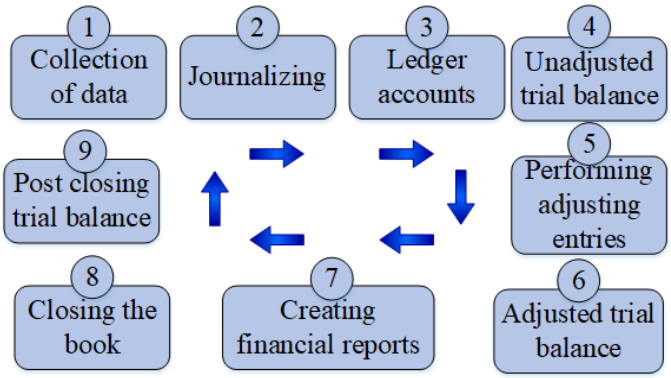



**Figure 1.** The cycle of accounting system.

### *3.2. Colored Petri Net*

Colored Petri net [2] or CPN is a mathematical modeling language for model checking [18]. It has been used for modeling and verifying the concurrent and distributed systems in various disciplines such as computer science, biology, and chemistry. CPN is combination of the discrete event dynamic system of classical Petri net and programming language, providing useful properties to allow the distinction between tokens. The core elements of a CPN model and PT model are alike. A place is an elliptical circle used for representation of a system state. A transition is a rectangle that represents an action or event of a system. An arc connects between a place and transition to show the direction of process flows. The state change is represented by the token's movement from a place to another place(s) by using the arc(s) crossing the transition. This action is called "transition firing". In the CPN modeling tool, a color set and variable declaration including an inscription expression

must conform to the syntax and lexicon of the back-end programming language of the modeling tool used.

Figure 2a shows the core CPN elements, and Figure 2b shows an example of the CPN model of the simple protocol that represents six data packages that are transmitted from the sender to the receiver through the network protocol. The transition "Send Packet" is enabled and it will fire one by one token to the place "A" , these events represent that the sender sends a message through the network protocol. Next, the transition "Transmit Packet" will fire a token to the place "B", demonstrating the message transmission to the receiver by the network protocol. Last, the transition "Receive Packet" consumes a token from the place "B" and fires two tokens to the place "Packets Received" and "C", displaying that the receiver gets the message. The receiver will acknowledge the sender via the network protocol with the transition firing of the transition "Transmit Act". However, this CPN model does not consider the message ordering, re-transmitting, and lost message.

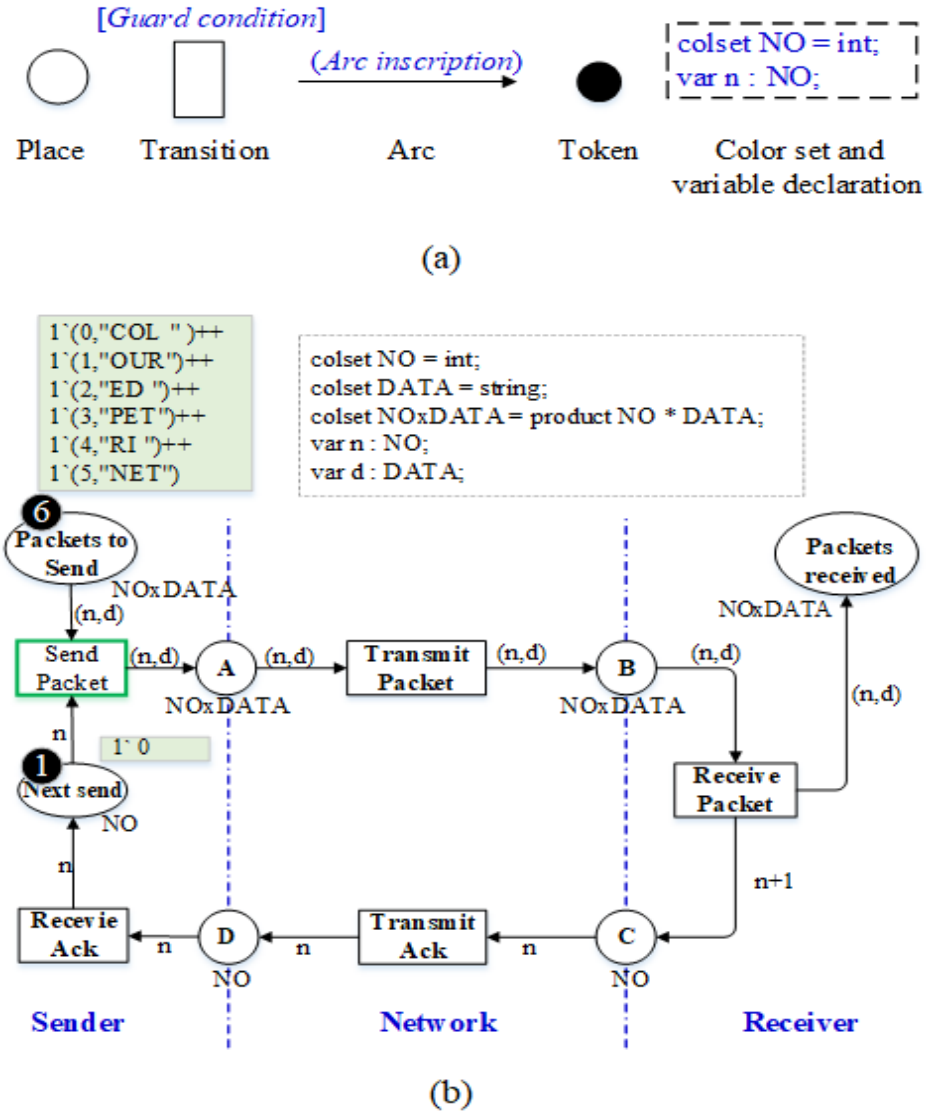



**Figure 2.** (**a**) Core elements of CPN. (**b**) Example of CPN model of a simple protocol.

CPN modeling tools usually provide a simulation mode and verification mode. The simulation mode works under a particular scenario, whereas the verification mode works on all possible states. To analyze a state space in verification mode, the state space generator and temporal logic [19] are provided. However, the state space explosion problem [20]

remains a drawback of verification mode if the CPN model has behaviors that are large in size or complex.

There are many works [21–25], not only in computer science, that use Petri nets for describing and simulating a system's biology and chemical reactions. It allows scientists to design and analyze the biological system network, acid cycle, metabolic pathway, and system behaviors. However, a classical Petri net cannot describe the low-level system behaviors. Thus, height-level Petri nets such as colored qualitative Petri net (QPNC), colored stochastic Petri net (SPNC), and colored continuous Petri net have recently applied to model and analyze biological systems.

## 4. Designing the Accounting System Using Colored Petri Net

This section details the CPN design model for instruction of the computerized accounting system. Our reason for choosing CPN is that the CPN's graphical representation allows us to show the data objects and data flows in an accounting system. It enables learners to trace back the data objects and system executions step-by-step. CPN tools also provide the analyzing perspective in both simulation mode and verification mode. We designed the CPN accounting system model according to the seven mentioned learning modules that are detailed in Section 3.1. Figure 3 shows the accounting modules and data flows of the CPN accounting model. We determined the contributions of the CPN accounting system for the following purposes.

1. To demonstrate the accounting system, accounting components and accounting modules in terms of the graphical model advocating the instruction of computerized accounting system;
2. To simulate the accounting behaviors and to verify the correctness of the accounting transaction flows.

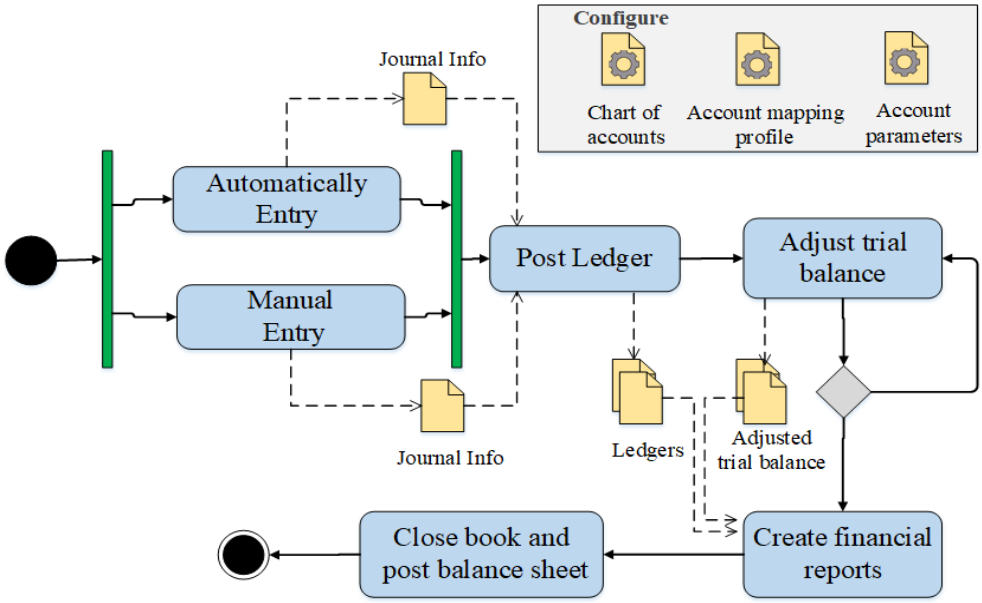



**Figure 3.** Accounting modules and flows of CPN model.

### *4.1. Formal Definitions of CPN Model and Accounting System*

We defined the formal definitions of CPN models and accounting systems to describe their elements and to illustrate the relations and constraints between the CPN model and the accounting system model as follows.

**Definition 1** (CPN model [2]). *A CPN model is a tuple CPN = (P, T, A, Σ, V, fs, fg, fe, fi) where:*

*P is a set of places representing a state of system.*

*T is a set of transitions representing an action of system, $P \cap T = \emptyset$ .*
*A is a set of arcs connecting between the place and transition, $A \subseteq ((P \times T) \cup (T \times P))$.*
*$\Sigma$ is finite set of color sets or data types.*
*V is a finite set of colored variables, $Type(v) \in \Sigma$, for all variables $v \in V$.*
*fs is a function used to assign a color set to each place, fs: $P \rightarrow \Sigma$.*
*fg is a guard-labeling function assigning a guard condition to a transition $t \in T$, fg: $T \rightarrow$ conditional expression and Type(fg(t)) = Boolean expression.*
*fe is a function used to assign an arc expression for an arc $a \in A$, fe: $A \rightarrow$ arc expression and Type(fe(a)) = fs(p) where p is the target place of arc a. It indicates that the color set of the target place must conform to the output data type of the arc expression. The arc expression may be a list of variables including the conditional expression that come along with a list of variables.*
*fi is an initialization function used to assign a marking to each place, fi:*
*$P \rightarrow$ initial expression, Type(fi(p)) = fs(p) for all the places $p \in P$. It indicates that the data type of the initial marking in a place must be conformed to the place's color.*

**Definition 2** (Accounting system)**.** *An accounting system is a tuple AC = (Ca, Aa, Ac, Bb, Jn, Jr, Ld, fAp, Lr, fMax, SumTotB) where:*

*Ca is a set of core account types. For $ca_i \in Ca$, $ca_i$ = (Name, Prefix, Type), where $ca_i$.Name indicates the core account name of $ca_i$ described as a string expression. $ca_i$.Prefix indicates the account code prefix of $ca_i$. $ca_i$.Type indicates the account type of $ca_i$. There are five core account types: Asset, Liability, Equity, Revenue, and Expense.*
*Aa is a set of accounts. For $aa_i \in Aa$, $aa_i$ = (Name, Code), where $aa_i$.Name indicates the account name of $aa_i$, and $aa_i$.Code indicates the account code of $aa_i$ described as a string expression, and the length of the account code depends on a system configuration.*
*Ac is a chart of accounts, which is a list of accounts where ($Ac \subseteq Aa$).*
*Bb is a set of beginning balances. For $bb_i \in Bb$, $bb_i$ = (Code, Bal, Year), where $bb_i$.Code indicates the account code of $bb_i$, which can be referred to the account code $aa_i \in Aa$ as well. $bb_i$.Bal means the total balance of $bb_i$ at the specific account period (month), and $bb_i$.Year is the year of an account period.*
*Jn is a set of journals. For $jn_i \in Jn$, $jn_i$ = (Date, Src, Vouc), where $jn_i$.Date is the date of journal. $jn_i$.Src determines the data source of the financial transaction used for generating the accounting transactions, and $jn_i$.Vouc indicates the voucher number of journal.*
*Jr is a set of accounting transactions in the journal. For $jr_i \in Jr$,*
*$jr_i$ = (Seq, Acc, RefTrx, TType, Amt), where $jr_i$.Seq is an ordering number of $jr_i$. $jr_i$.Acc indicates the account code. $jr_i$.RefTrx indicates the reference number of the financial transaction of $jr_i$, whereas $jr_i$.TType and $jr_i$.Amt indicate the T-account entry type (Dr or Cr) and the T-account amount, respectively.*
*Ld is a set of ledgers, $Ld \subseteq Aa$. For $ld_i \in Ld$, $ld_i$ = (Name, Code), where $ld_i$.Name is the name of the ledger. $ld_i$.Code is the account code that refers to the account code declared in the chart of accounts ($ld_i$.Code = $aa_i$.Code).*
*fAp is a mapping function used to indicate the core account type, fAp: $Aa \rightarrow ca$, or fAp: $Ld \rightarrow ca$.*
*Lr is a set of accounting transactions in the ledger, for $lr_i \in Lr$, $lr_i$ = (ID, Date, Type, TotalBl), where $lr_i$.ID is an ordering number of $lr_i$; $lr_i$.Date indicates the transaction date of $lr_i$. $lr_i$.Type is the ledger transaction type composing the Beginning and Accumulating; and $lr_i$.TotalBl indicates the net balance of $lr_i$ .*
*fLd is a mapping function used to indicate the ledger of the accounting transaction, fLd: $Lr \rightarrow Ld$.*
*fMax is a function used to retrieve the last number of the accounting transaction in the ledger. fMax: $Lr \rightarrow \mathbb{N}$.*
*fList is a binding function used to retrieve the last accounting transactions of all ledgers under a specific core account type, fList:Ca $\rightarrow$ a list of accounts, where $lr_i \in Lr$ comes from the last transaction in each ledger.*
*fSumTB is a function used to summarize the total balance of all ledgers, grouped by a specific core account type, fSumTB:Ca $\rightarrow \sum_{i=1}^{n} \equiv fList(Ca).TotalBl_i$.*

### 4.2. Modules of CPN Accounting System

As the accounting modules and data flows shown in Figure 3, the modules of CPN accounting model are designed by using the CPN tools as the seven CPN modules or CPN constructs as follows.

1. Configuration module
2. Journal entries module
3. Ledger account module
4. Lost/Profit module
5. Trial balance module
6. Balance sheet module
7. Integrated module (all the above modules are integrated to form an integrated module, and it is arranged in hierarchical structure.)

The lost/profit module, trial balance module and balance sheet module are in the "Create financial reports" part shown in Figure 3. The process of a book closing and balance sheet posting is an updating and flags the account status to be "closed"; thus it is designed as a CPN function. As the mentioned modules, we designed the features of the CPN constructs follows:

1. All the modules can be simulated and verified by using the simulation palate of the CPN tools for checking and tracing the CPN model execution.
2. Provide the journal entry scenarios or case studies of the two business domains—a retail system and mortgage loan system—that come along with the accounting case studies for simulating the journal entries. The journal entry scenarios are defined in terms of a set of the token colors. The learners can take them to initiate and simulate to see the accounting system behaviors.
3. The learners can use the provided CPN function to collect all the model execution events occurring during a simulation or verification and can export the execution events as a log file if they need.

The details of each module are described in the Sections 4.2.1–4.2.6. The CPN constructs are designed by using a bottom-up approach, and we demonstrated how the CPN model helps the learners and constructors in the accounting system classes. Each CPN module comes along with an activity diagram presenting the process flows and data flows, and the learner can compare the task sequence and data flows in the activity diagram with those of the CPN construct as well. The CPN construct of each module is designed and tested individually; next, all the designed CPN modules are integrated accordingly to the accounting system topology in a topmost perspective. The integrated CPN model is used to test the transaction flows occurring at the general ledger posting state until the creation of the financial reports. Figure 4 shows the topmost CPN model arranged in a hierarchical structure.

#### 4.2.1. Configuration Module

We mimicked the configuration of accounting system to be a set of the CPN inscriptions named the Config-Net. The learners can modify the accounting configuration and simulate the CPN model based on the configured parameters. The Config-Net is used to determine:

1. The CPN model parameters (Param-Net) that comprise the configuration of the account period, running number of voucher, validating policies, and the beginning balances of ledger (Begin-Net).
2. A chart of accounts (Chart-Net).
3. An account mapping profile (Map-Net).

The color sets used in the CPN model are mapped from the primitive data types determined in the system specification (data dictionary) of a real accounting system, including the data structure of token color and a production color set is derived from a

list of attributes in the data dictionary. Figure 5 shows the model inscriptions and the CPN pallet used for simulating the CPN model. For instance, the declaration of a color set "DATA" in Figure 5a is a primitive data type of "String" that is mapped from Varchar (50) in the data dictionary. Figure 5c is the Param-Net that is the global variables used for the configuration of an account year, account month, and account period. The Chart-Net and Begin-Net are declared as a string arrayLists data type. The Map-Net is implemented as a CPN function that acts as a decision tree for an account code mapping. The learners can customize these CPN inscriptions arbitrarily. The CPN variables and CPN functions, including all CPN inscriptions in these configurations are used in the process of the journal entries, ledger posting, and financial report.

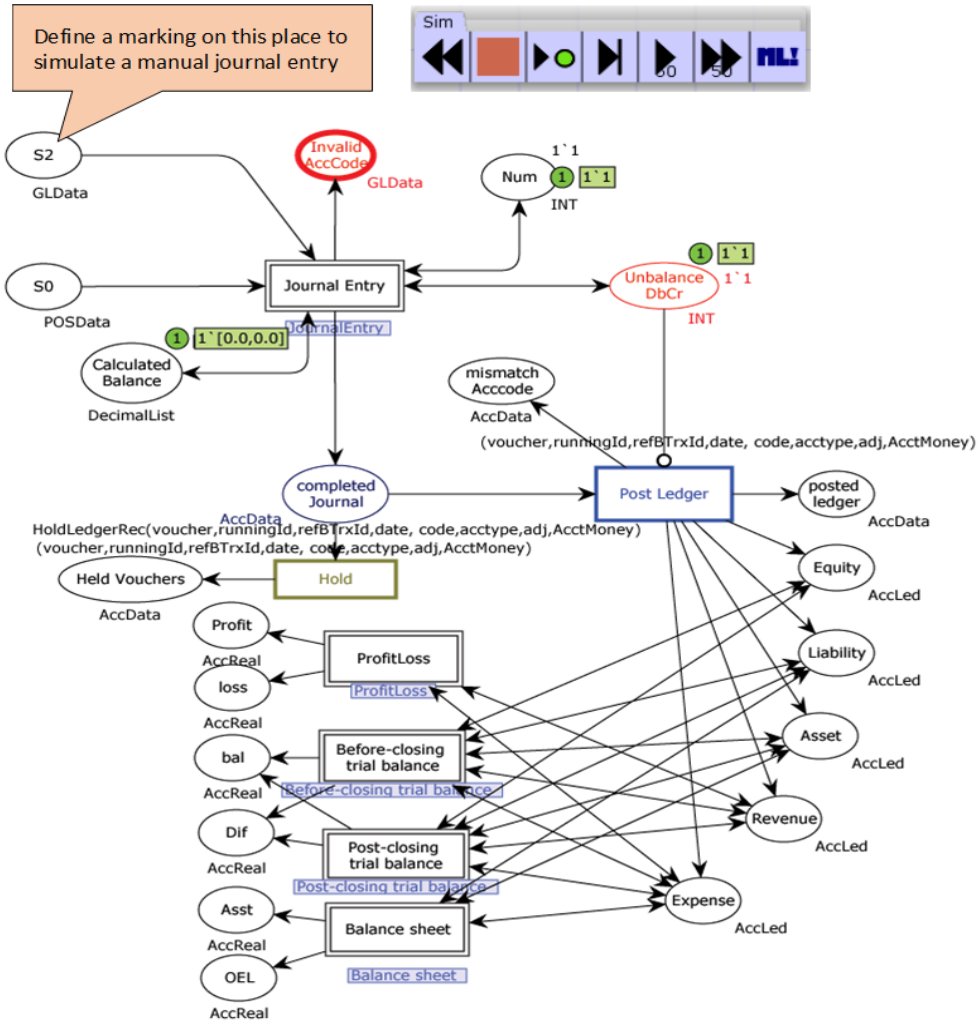



**Figure 4.** Topmost hierarchical CPN model: the double-line transitions are the substitute transitions representing the accounting modules. The green dot on a place is the number of tokens and the green box is the color set. For example, place "Calculated Balance" "(1) 1′ [0.00,0.00]" is that there is one token on it, and both Credit and Debit balance start with 0.00.

#### 4.2.2. Journal Entries Module

The CPN construct of the journal entry module is designed based on the two entry channels: (1) the manual entry and (2) the automatic entry that receives financial data from the point of sale module. The manual entry part supports the simulation of the accounting transactions derived from a consideration and endorsement by the accountants, whereas the automatic entry part represents the processing of accounting transactions sent from the point of sale module to GL module. Figure 6 shows the CPN construct of the journal entries.

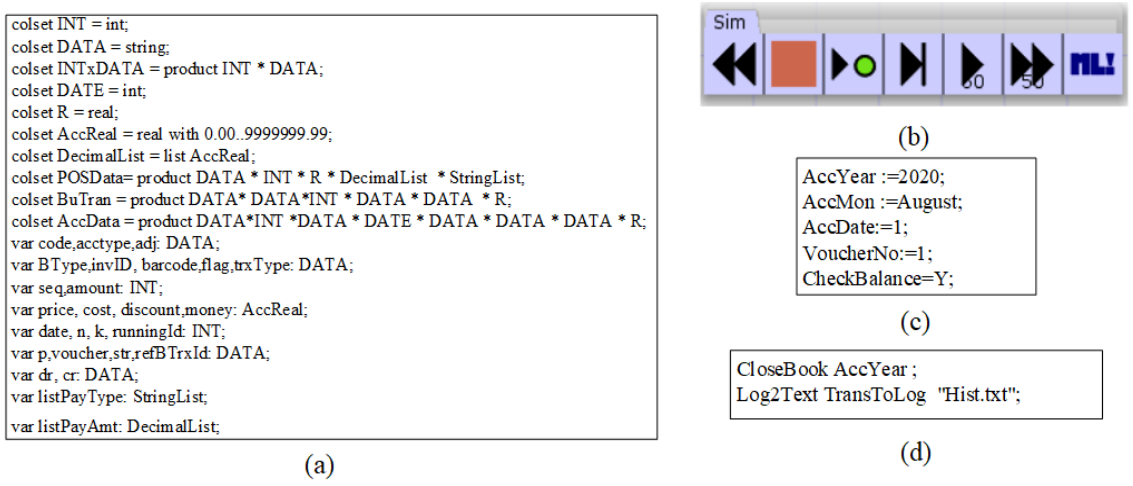



**Figure 5.** Inscriptions of the CPN model: (**a**) Excerpt Color sets, variables and functions of the model, (**b**) Simulation pallet, (**c**) Accounting parameters, (**d**) CPN inscriptions that are implemented as a function for closing the book and exporting the simulation log. The chart of account is declared as a string ArrayList data type and the account mapping profile is declared as the CPN functions in a part of (**a**).

Simulating the Manual Journal Entry

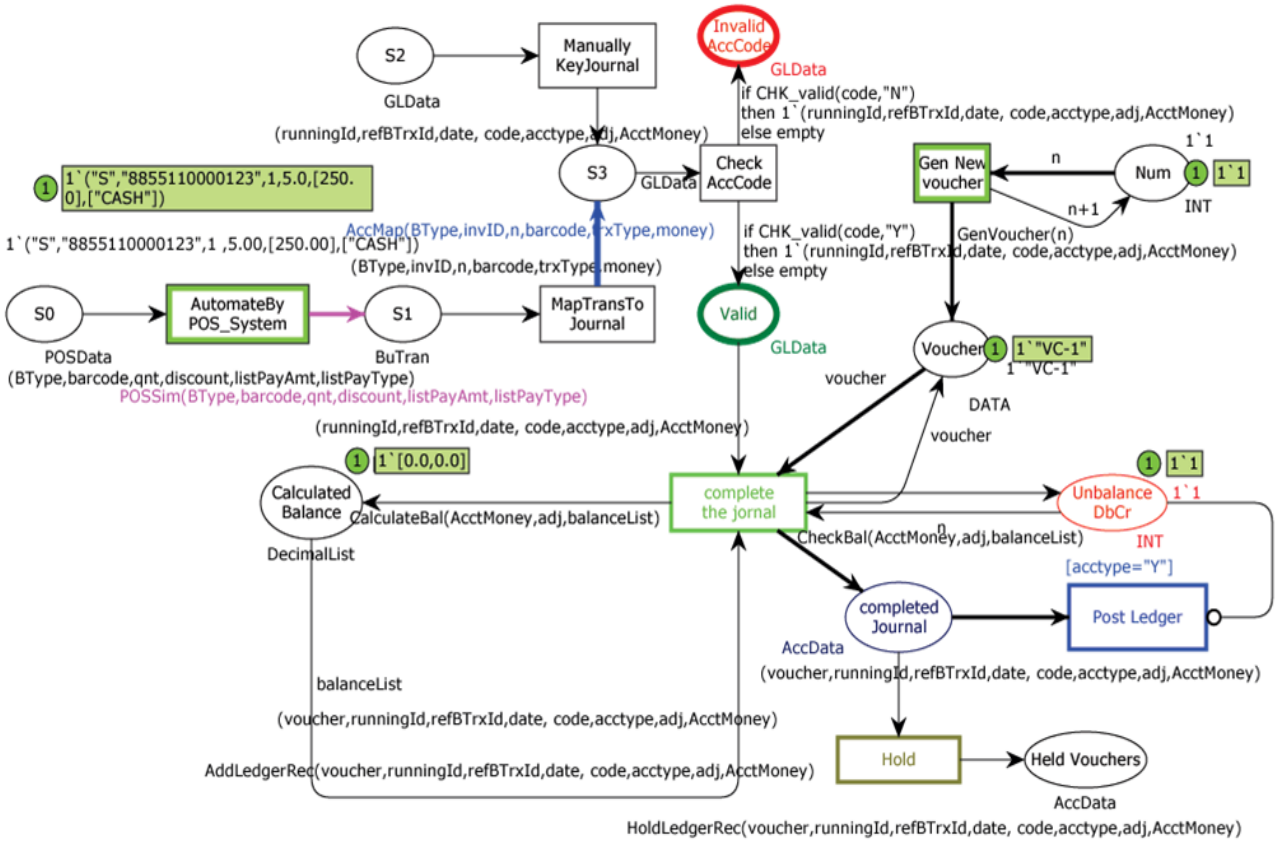
Considered the CPN construct in Figure 6a, the place "S2" is the place for determining the accounting input data with the date, the amount of the debit and credit. The marking of the manual journal entry differs from that of the automatic journal entry, and the marking in the place "S2" must be conform to the accounting data format of a general ledger database. The transition "ManuallyKeyIn" represents the scenario in which the accountants manually classify the raw transactions, mapping the raw transaction into the accounting transactions, and filling the derived accounting transactions into the journal. After the transition, "ManuallyKeyIn" consumes the token, the transition fires the tokens into the place "S3". We implemented the CPN function to capture and collect the actions and data flows as a simulation log. The learners can use it to trace the accounting process if they need to know what happen during a simulation or verification. Figure 6b illustrates the excerpt simulation log of the manual journal entry.

Simulating the Automatic Journal Entry

Considered the CPN model in Figure 6a, the learners can simulate the automatic journal entry by determining the token color in place "S0". In actual, this place connects to the output place of the POS module. The determined token color in this place represents the raw data that are received from the POS module. In the simulation process, the transition "AutomateByPOS_System" consumes the tokens in place "S0". The tokens will pass the arc inscription with the CPN function named "POSSim". The token colors are passed by a list of the variables: *barcode*, *amount*, *price*, and *discount*. Next, the tokens containing the output data will pass the transition "MapTransToJournal" and call the function "AccMap" in order to map the raw data into the accounting transactions. These processes are the activities sequence and data flows of a journal entry process that the learners should be comprehended.

For instance,the initial marking in the place "S0" with "1'("S","8855110000123", 1, 5.00, [250.00], ["CASH"])" is determined to simulate the scenario in which the cashier records the normal sale transaction with the product code "8855110000123", sold 1 item. The product price is 255 USD, and the cashier gives a 5 USD discount. The cashier receives the cash 250 USD. (Actually, the cashier may receive more than one payment type; thus, the payment types and the received amounts are recorded in an *Arraylist[]* data type.) The transition "AutomateByPOS_System" consumes and fires the token into place "S1" by using the CPN function "POSSim". Next, the transition "MapTransToLedger" consumes

the token in place "S1" one by one. The transition "MapTransToLedger" also produces the tokens into the place "S3" by using the function "AccMap" on the output arc. These simulation steps are provided for the learners to animate the processes regarding the transformation mapping of raw data into the accounting transactions. The learners can check a type of the financial transaction determined in the state "S1" (where a system code PP: product price, PC: product cost, DC: discount, CH: cash). In the case of the learners giving an incorrect input sequence or mismatched data type, the model will prompt an error message.




(a)

```
Mode:verification
Step    Action                Data objects (Token colors)
-------------------------------------------------------------------------------------------------
0       init                  1`("S","8855110000123",1 ,5.00,[250.00],["CASH"])
-------------------------------------------------------------------------------------------------
1       AutomateByPOS_System  1`("S","8855110000123",1 ,5.00,[250.00],["CASH"])
-------------------------------------------------------------------------------------------------
2       POSSim                1`("INV001",1, 8855110000123,"PP",255.00)++
                              1`("INV001",2, 8855110000123,"PC",170.00)++
                              1`("INV001",3, 8855110000123,"DC",5.00)++
                              1`("INV001",4, 8855110000123,"CH",250.00);
-------------------------------------------------------------------------------------------------
4       MapTransToJournal     1`(1,"Voucher001",20200801,"10002","Cash on hand","DR",250.00)++
                              1`(1,"Voucher001",20200801,"50001","Product cost","DR",170.00)++
                              1`(1,"Voucher001",20200801,"50002","Sales discount","DR",5.00)++
                              1`(1,"Voucher001",20200801,"12002","Dep. goods inventory","CR",170.00)++
                              1`(1,"Voucher001",20200801,"40001","Sales","CR",255.00);
-------------------------------------------------------------------------------------------------
```

(b)

**Figure 6.** (**a**) The CPN constructs of the journal entries. (**b**) Excerpt log of journal entry.

To demonstrate the validation process of accounting transaction system, the accounting transactions that come from the manual journal entry and automatic journal entry will be re-checked by the transition "CheckAccCode" to verify whether the account code of each transaction is valid or not. If the account code exists in a chart of accounts, the token will appear in the place "Valid". In contrast, the mismatched account code will be removed, and the token will appear in place "Invalid AccCode". In the case of a valid transaction, the transition "complete the journal" consumes the valid accounting transaction. The transition consumes the tokens accordingly to the sequence number (running number). Next, the net balance calculation and the voucher number determination are performed. The net balance is calculated, and the accounting transactions are flagged to be "completed" simultaneously. In the place "completed journal", the accounting transactions in each voucher can be posted into the ledgers if and only if the net balance of voucher is equal. The net balance checking function (CalculateBal on the output arc of transition "complete the journal") is implemented for checking the total balance of Dr and Cr of each voucher. The learners can disable the net balance checking function by changing the configuration parameter named "CheckBalance" in the Config-Net to be "0" (see Figure 5c). The formula used for checking the equivalence of the net balance of each voucher is expressed as Equation (1).

$$\delta_i = \sum_{j=1}^{n} Dr_j - \sum_{k=1}^{n} Cr_k \quad (1)$$

The different values of *Dr* and *Cr* of each voucher *i* is $\delta_i$, calculated where the *i* is a voucher or an invoice in which all the accounting transactions *j* mapped by considering the account code and the *Dr* or *Cr* . If the $\mid \delta_i \mid$ is 0, the values of *Dr* and *Cr* of the voucher *i* is balance, while another case is unbalance.

#### 4.2.3. Ledger Posting Module

The ledger posting is a process in which the learners gradually record the accounting transactions classified in the journals into the ledgers. In a back-end process of CPN model, the system posts the accounting data into the corresponding ledger relying on the account code of each accounting transaction. This process is performed after the learners click on the transition "Post Ledger" while the place "Completed Journal" occupies more than one marking. The system records data into the ledger and makes the reference number be between the journal and ledger. Actually, the account type and corresponding account code are assigned for every transaction in the journal already. Thus, the ledger posting can be proceeded in a few minutes. We design the ledger posting process as the CPN construct shown in Figure 7.

The transition "Post Ledger" consumes the tokens in the place "completed journal" if and only if the net balances of voucher is equal. The accounting transactions will be posted into the ledgers, ordered by the voucher number and the sequence number of accounting transaction. Since a core account has more than one sub account code and ledger, we design the structure of the ledgers classified according to the five core account types. For instance, the place Asset contains the ledgers under the sub-account Asset(they are determined by the account code prefix "1"). The sequence number 0 and reference code "BB" identifies the beginning balance of ledger, which the net balance comes from the Begin-net. Another is the accounting transactions that come from the ledger postings. the sequence number 1 is the ledger that the net balance comes from the historical accounting transactions. The accounting transactions posted are collected in the place "posted ledger".

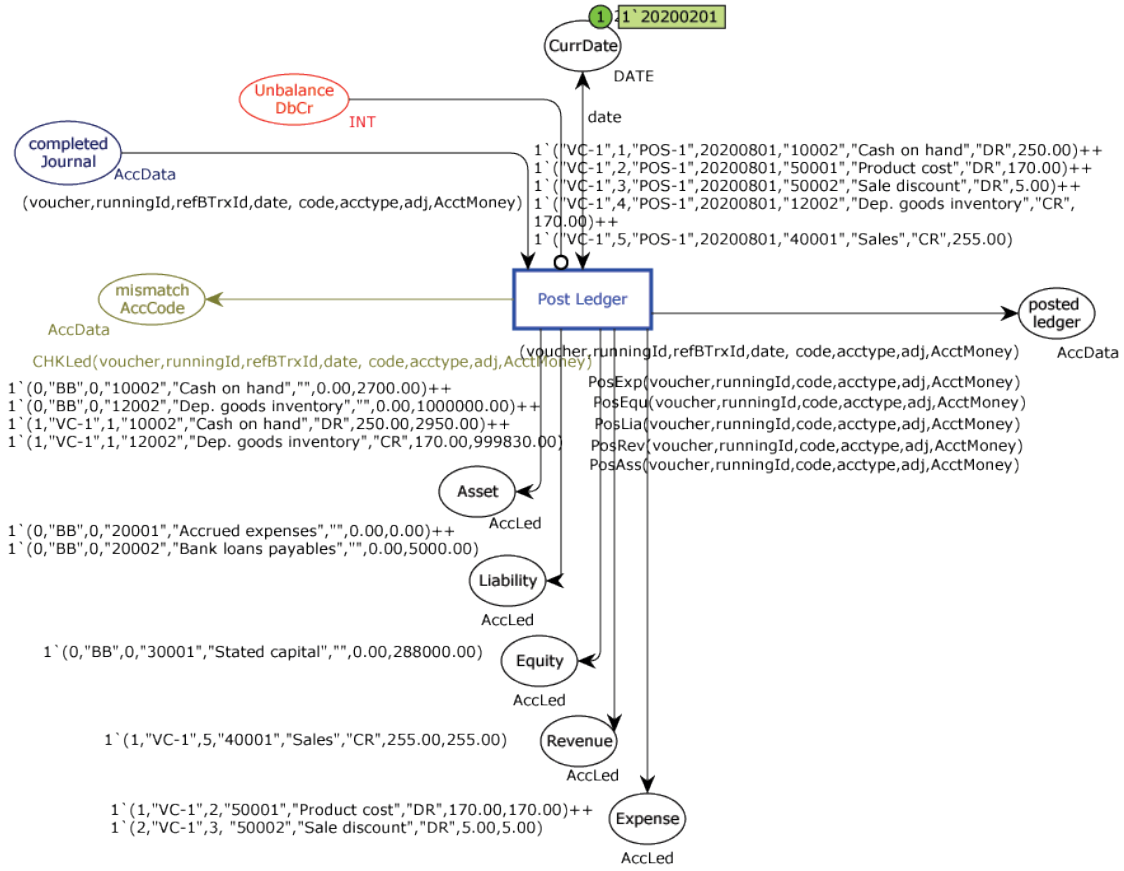



**Figure 7.** The CPN construct of the ledger posting.

#### 4.2.4. Lost/Profit Module

There are three core financial statements for checking the net balances of Dr and Cr and generating the financial reports of the organization. The function *fList()* and *fSumTB()* are used to create the financial statements. The CPN constructs of the financial reports creation are shown in Figures 8–11. The details of each financial report are described as follows:

**(1) The profit–loss report:** it is a primary report of accounting system, showing the total income and total expenses over a specific account period. The profit–loss report is calculated by introducing the parameters into Equation (2).

$$PL_r = \sum_{i=1}^{n} Rev_i - \sum_{j=1}^{n} Exp_j \tag{2}$$

*PL* is the profit or loss report, and *r* is the account period. The $Rev_i$ is the total balance of the ledger *i* under the core account *Revenue*. Furthermore, the $Exp_j$ is the total balance of the ledger under the core account *Expense*. In the case of a detailed report showing a list of accounts, the list of accounts under the core account *Revenue* and *Expense* are retrieved by using the function *fList()* declared in the Definition 2. Figure 8 shows the CPN construct of the profit or loss report.

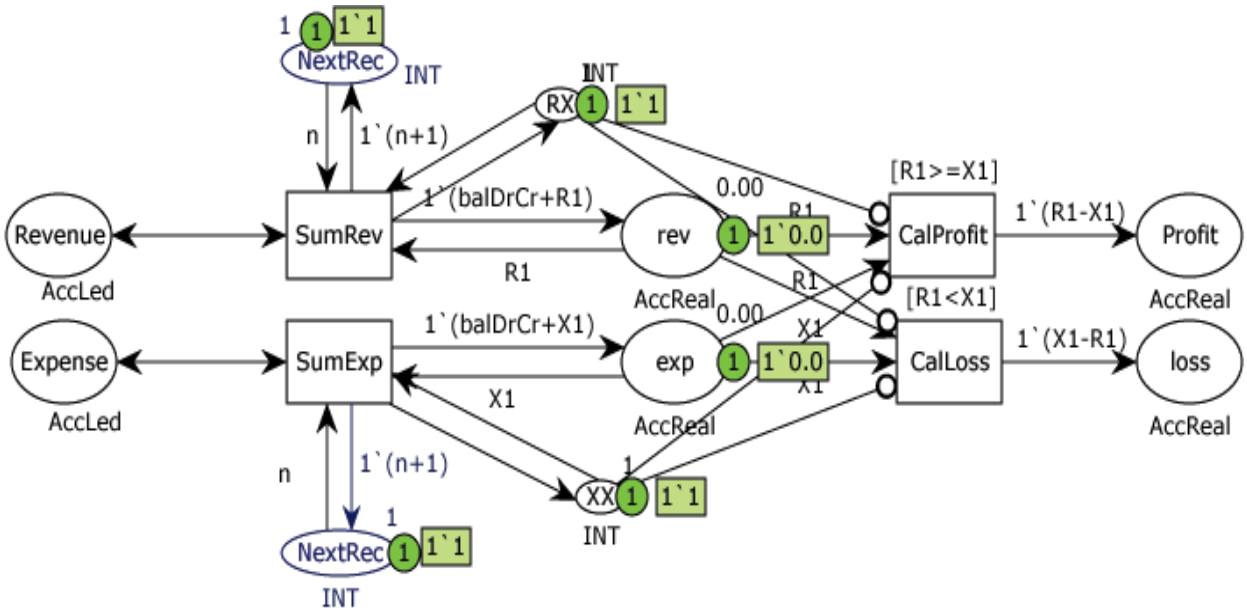



**Figure 8.** The CPN construct of the profit-loss report.

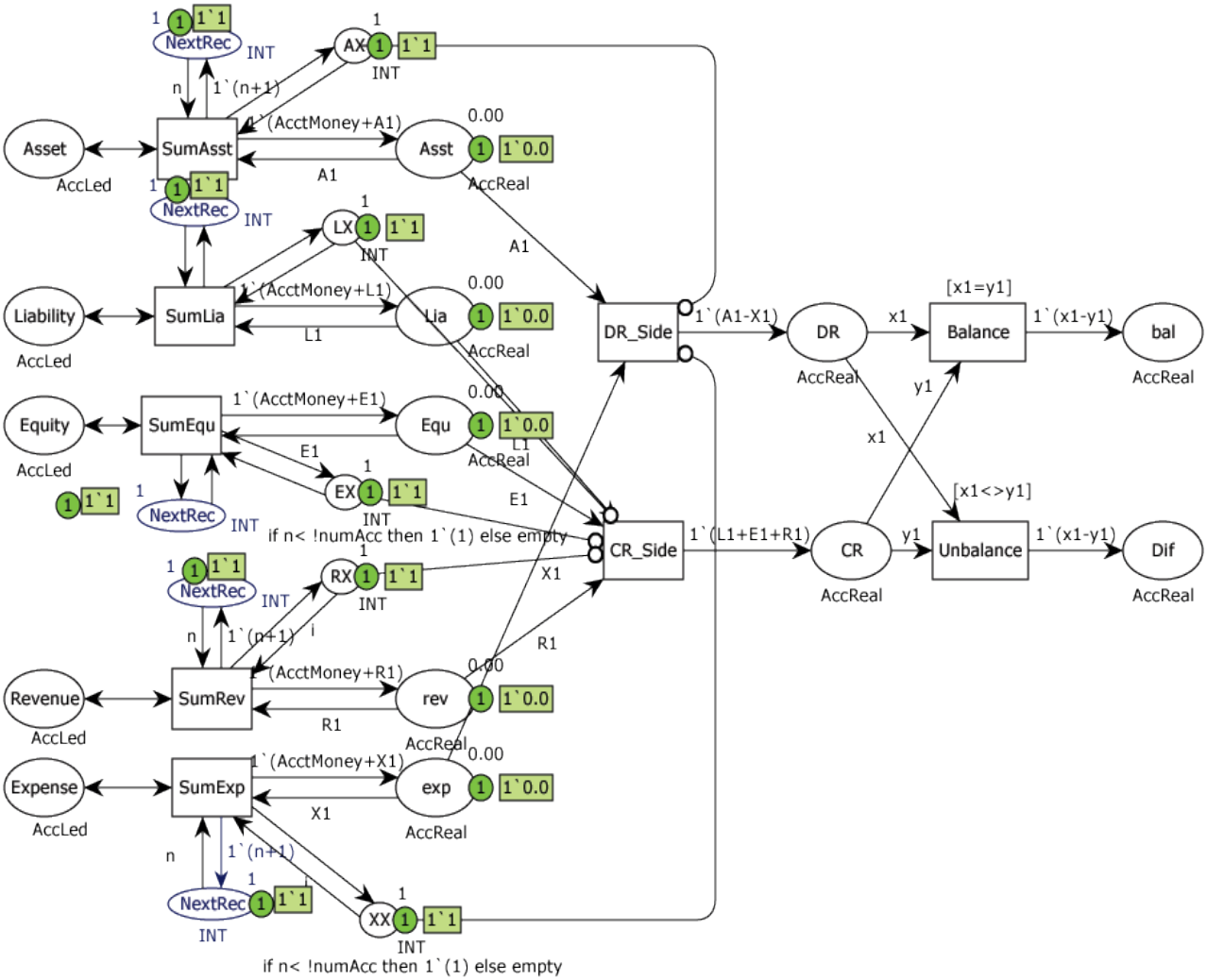



**Figure 9.** The CPN construct of the before-closing trial balance report.

#### 4.2.5. Trial Balance Module

The reports are for internal use to check the bookkeeping. The trial balance report can be made at before or after the account closing. For the before-closing trial balance report, the summation of the net balances of the account *Asset* and *Expense* must eventually be equal to that of the account *Liability*, *Equity*, and *Revenue*. Furthermore, the part of the account *Equity* does not include the profit or loss of current account period. The post-closing trial balance report is calculated based on the same formula but the retained earnings on the account *Equity* include the profit or loss of current account period already. The equations of the trial balance reports are typically written as Equations (3)–(7). The CPN constructs of the before-closing trial balance and post-closing trial balance reports are illustrated in Figures 9 and 10, respectively.

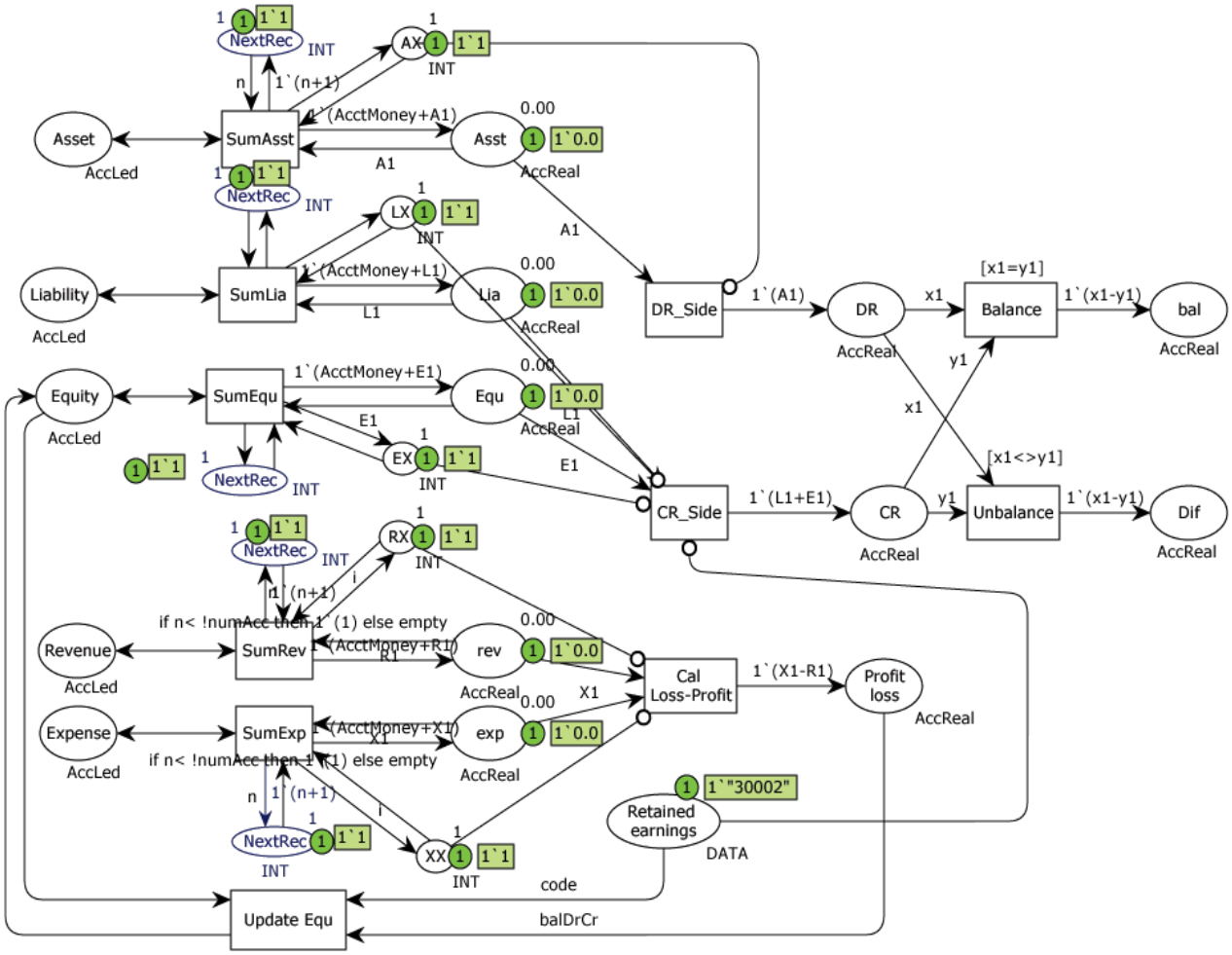



**Figure 10.** The CPN construct of the post-closing trial balance report.

There are two parts of the net balances, which *Dr* and *Cr* are grouped by the nature of account. The *Dr* is the summation of the net balances of the ledgers under the core account *Asset* and *Expense*. While, the *Cr* is the summation of the net balances of the ledgers under the core account *Liability*, *Equity*, and *Revenue*. In the case of the post-trial balance report, the *Cr'* is calculated based on the equities that include the profit–loss $PL_r$ already. $TB_b$ is the before-closing trial balance and $TB_p$ is the post-closing trial balance report.

$$Dr = \sum_{i=1}^{n} Asst_i + \sum_{j=1}^{n} Exp_j \tag{3}$$

$$Cr = \sum_{k=1}^{n} Lia_k + \sum_{l=1}^{n} Equ_l + \sum_{m=1}^{n} Rev_m \tag{4}$$

$$Cr' = \sum_{k=1}^{n} Lia_k + PL_r \tag{5}$$

$$TB_b = Dr \cup Cr \tag{6}$$

$$TB_p = Dr \cup Cr' \tag{7}$$

4.2.6. The Balance Sheet Report

The balance sheet report presents the organization's assets, liabilities, and equities. The net balances are calculated by using the parameters in Equations (8)–(10). The balance sheet report $TB_b$ contains the net balances of the account *Asset* and the summation of the liabilities including the shareholders' equities. If the accounting transactions are journalized and posted correctly, the net balances of *Asset* and *LiaEqu* will be equal. Figure 11 shows the CPN construct of the balance sheet report.

$$Asset = \sum_{i=1}^{n} Asst_i \tag{8}$$

$$LiaEqu = \sum_{i=1}^{n} Lia_i + \sum_{j=1}^{n} Equ_j \tag{9}$$

$$TB_b = Asset \cup LiaEqu \tag{10}$$

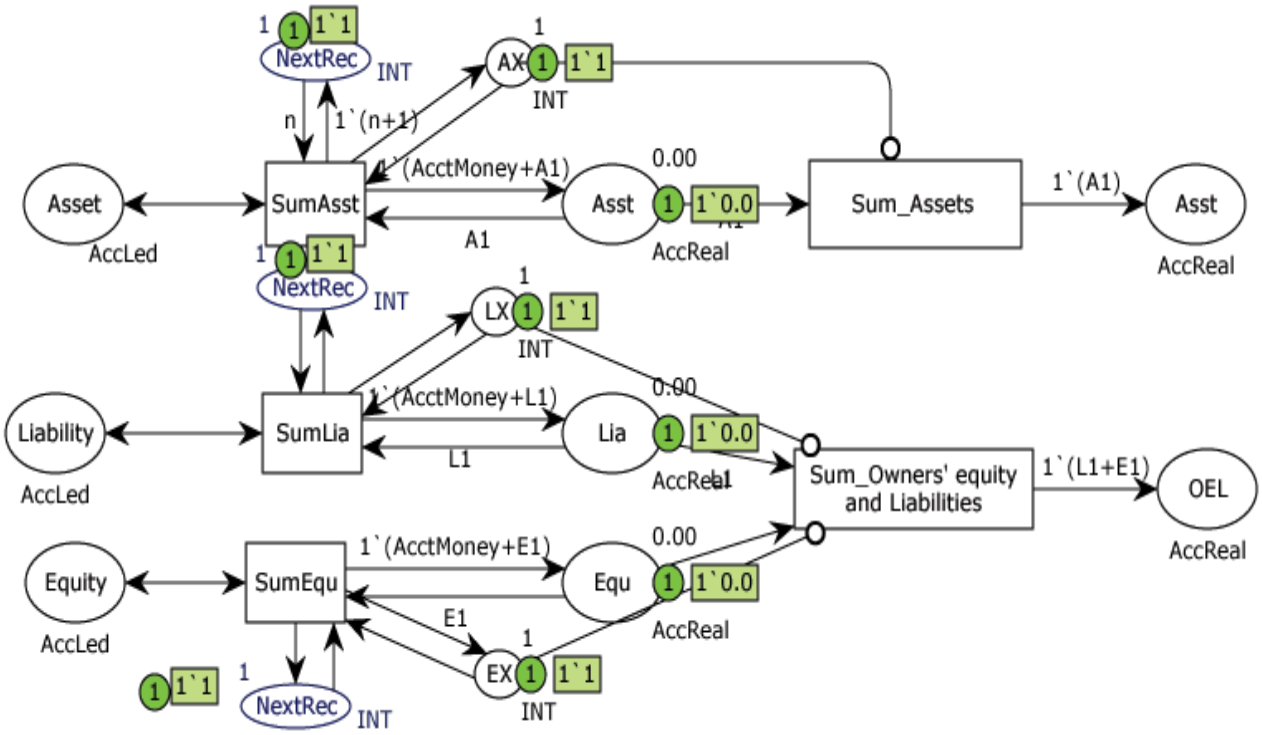



**Figure 11.** The CPN construct of the balance sheet report.

## 5. Evaluation the CPN Accounting System Model

To analyze the proposed CPN constructs and to provide the case studies for the learners, each module is tested using the simulation mode. All the CPN constructs are integrated to be a whole accounting system model. The integrated CPN construct can be represented in a flat and hierarchical structure. The flat model is sophisticated because of the model is huge in size and includes complex events. The hierarchical CPN model is compact, reducing the model complexities and hiding some information. The topmost CPN model of the accounting system is shown in Figure 4. This model is arranged in a hierarchical structure in which the CPN constructs are determined as the CPN substitute transitions. It is an integration of the CPN constructs and inscriptions obtained from the Sections 4.2.1–4.2.6.

We evaluated the proposed CPN model by validating the accounting processes based on the ground truth data of the retail department store (POS) and the mortgage loan system (LOAN). The CPN model is customized by re-configuring the Config-Net, based on the the ground-truth data of each system. We validated the model in both the simulation mode and verification mode. The number of the accounts, transaction types, and test cases used are described in Table 1. The test cases are also provided in terms of the CPN markings, which the learner can choose to simulate arbitrarily. The generation of test cases and the initial markings determination are an application of the decision-table-based testing [26] to represent the possible cases of journal entries. We generate forty-nine makings or test cases, calculated from the seven conditions of the accounting transaction types (A0–A6), the two conditions of the net balance (P0, P1), the two conditions of the account matching status (C0, C1), and the two conditions of the journal holding (H0, H1). The forty-nine cases are derived from a production ($7 \times 2 \times 2 \times 2$), from which 7 cases are deducted (the duplicating cases). Table 2 shows the equivalence classes used for generating the test cases and initial markings. The excerpt of the decision table of the POS system is in Table 3.

**Table 1.** The number of the account codes, transaction types, and test cases used in our experiments.

| Accounting Properties and Test Cases | POS | LOAN |
|---|---|---|
| Account codes and ledgers | 57 | 77 |
| Transaction types: (Auto entries) | 6 | 7 |
| Test cases: (Auto entries) | 49 | 57 |
| Transaction types: (Manual entries) | 5 | 8 |
| Test cases: (Manual entries) | 41 | 65 |

**Table 2.** The equivalence classes used to guide the learners to create the simulating cases.

| Conditions | Types |
|---|---|
| A0 = bType : XX | Invalid |
| A1 = bType : NS (Normal sale) | Valid |
| A2 = bType : DS (Deposit sale) | Valid |
| A3 = bType : VS (Voided sale) | Valid |
| A4 = bType : VD (Voided deposit sale) | Valid |
| A5 = bType : RS (Returned sale) | Valid |
| A6 = bType : RD (Returned deposit sale) | Valid |
| P0 = Dr $\neq$ Cr | Valid |
| P1 = Dr=Cr | Valid |
| C0 = Matched code | Invalid |
| C1= Mismatched code | Valid |
| H0= Holding journal | Valid |
| H1= Unholding journal | Valid |

**Table 3.** The excerpt of the decision table used to create the test cases of POS system.

| **Condition** | **Cases** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **1** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **...** | **49** |
| Business transaction type | A0 | A1 | A1 | A1 | A1 | A1 | A1 | A1 | ... | A6 |
| The net balance of Dr and Cr of a voucher | P0-P1 | P0 | P0 | P0 | P0 | P1 | P1 | P1 | ... | P1 |
| Account code matching | C0-C1 | C0 | C0 | C1 | C1 | C0 | C0 | C1 | ... | C1 |
| Journal holding | H0-H1 | H0 | H1 | H0 | H1 | H0 | H1 | H0 | ... | H1 |
| Rule Count | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | ... | 1 |
| **Action or State(CPN Place )** | | | | | | | | | | |
| Generate financial transactions (“S1”) | | x | x | x | x | x | x | x | ... | x |
| Invalid account code (“Invalid AccCode”) | | | | x | x | | | x | ... | x |
| Valid account code (“Valid”) | | x | x | | | x | x | | | |
| Dr = Cr (“Calculated Balance”) | | | | | | x | x | x | ... | x |
| Dr $\neq$ Cr (“Calculated Balance”) | | x | x | x | x | | | | | |
| Unbalance (“UnbalanceDrCr”) | | x | x | x | x | | | | | |
| Held journal (“Hold Vouchers”) | | x | | x | | x | | x | | |
| Completed journal | | | | | | x | x | | | |
| Post the ledgers | | | | | | x | x | | | |
| Posted ledger | | | | | | x | x | | | |
| The function “POSSIM” throws an error message | x | | | | | | | | | |

The examples of the markings and the test scenarios of model simulation are as follows:

Marking No. 1: (refer to case no. 1 in Table 3).

“1‘(“XX”,“8855110000123”, 1, 5.00, [250.00], [“CASH”])“, this marking simulates the scenario in which the account type of the financial transaction is invalid. It is determined by “XX”, where the bType “XX” does not exist in the Chart-Net. The expected result of this case is that the function POSSIM must show an error message.

Marking No. 2: (refer to case no. 9 in Table 3).

“1‘(“NS”,“8855110000123”, 1, 5.00, [250.00], [“CASH”])”, this marking is to simulate the scenario in which the CPN model can automate the valid accounting transactions but the net balance of Dr and Cr is unbalanced. Faults of the incorrect account code mapping must be fed into the Map-Net before a simulation. In a simulation stage, the model produces a set of the accounting transactions conforming to the account mapping profile but the net balance is unequal. This simulation will reach the state “UnbalanceDrCr” and possible to reach state “Held Voucher” .

As the information in Table 1, we simulated the CPN model with 90 markings or test cases (49 + 41) for the POS system and with 122 markings (57 + 65) for the LOAN system. The model simulates the accounting transaction flows from the journal entry stage until the financial report stage. We also validate the CPN model by using the verification mode to find deadlock and live-lock, based on the the ground truth data and Config-Net of the POS system. Due to the huge size of the CPN model, the state space generation is time consuming for the verification mode. The state space graph contains 9,781,269 nodes and 8,563,775 edges, and its generation process is the time-consuming. Therefore, the verification mode is suitable for verifying the CPN module or the sub-process individually. Furthermore, it should be used together with the partial verification techniques in order to reduce the time consumption and avoid the state space explosion problem. The verification mode is suitable for learners who have experiences in model checking.

The results of the experiments of both the simulation mode and verification mode show that the CPN model can represent the accounting transaction flows from the journal entries stage until the financial-report-generation stage correctly and consistently. The CPN tools can simulate all possible cases of journal entries; it can also identify the faults and deadlocks in the CPN model when the learners determine incorrect markings. The CPN model has been used in the accounting system classes for the two learner groups: bachelor

degree in accountancy and bachelor degree in computer information system for 30 and 60 persons, respectively. We received a satisfactory report that the learners in a major of computer information system were satisfied, while the learners in the accountancy major were moderately satisfied. They gave as the reasons that the model enlarges the better accounting term, but the user interface of the CPN tools quite is difficult to use for practising. This indicates that the proposed CPN model advocates the educational perspective for who are studying in the accounting system, journal entries, and accounting transaction flows. However, it should be enhanced regarding the user interface in the future.

Table 4 shows the comparison of related works and our framework, which the capabilities of each framework are separated into three parts: model, accounting feature and formal verification. The framework of Poolsawasdi and Dechsupa [6] is CPN-based model automated from the BPMN models. Although the CPN model is created by using CPN tools that support model simulation and parameterization, the authors did not consider the execution log exporting and accounting parameter configuring. The accounting behaviors simulation partially supports the ledger system and account mapping profile because the model is designed for checking the correctness properties in low-level design. While our approach extends the capabilities of the accounting behaviors simulation, execution log exporting, including accounting parameters configuring that is an important feature for the accounting system instruction. The framework of Kim et al. [7]. is created by using classical Petri net language relied on the business process models and user requirements, designed for the capacity and performance checking. This work does not encourage the accounting behaviors simulation and qualitative verification but it contribute to the performance analysis of accounting workflow in hight-level design.

As the two contributions mentioned in Section 4, we sum up our approaches as follows.

1. To demonstrate the accounting system that advocates the instruction of the accounting system and journal entries, we provided the CPN accounting system model that supports the simulation mode and verification mode. The obtained CPN model represents the accounting transaction flows likely to occur in an actual accounting system. It comes along with 232 cases of the journal entries for the retail system and loan system.
2. To simulate and verify the correctness of the accounting transaction flows, we provided the CPN modules that come along with the CPN function that allows the learners to trace the model's execution as a system log. Furthermore, the learners can configure a chart of accounts, account mapping profile, and the beginning balance, including the simulation of journal entries by using the simulation palate of CPN tools.

**Table 4.** Comparison of contributions in related works and our work.

| | **Poolsawasdi and Dechsupa [6]** | **Kim et al. [7]** | **Our Approach** |
|---|---|---|---|
| **Model** | | | |
| Language | CPN | PT | CPN |
| Creation | Automate form BPMN model | Manual | Manual |
| Simulation | Partial | Yes | Yes |
| Execution log | No | No | Yes |
| **Accounting features** | | | |
| Parameterization | No | No | Yes |
| Ledger system | Partial | No | Yes |
| Journal entry | No | No | Yes |
| Account mapping profile | Partial | No | Yes |
| financial report | No | No | Yes |
| **Formal verification** | | | |
| Level of abstraction | Low | Hight | Low |
| Properties | Correctness | Capacity and performance checking | Correctness |
| **Appropriate for instruction** | No | No | Yes |

## 6. Conclusions and Future Work

The instruction of a computerized accounting system and general-ledger accounts posting is a part of the accounting information system. We proposed the application of a CPN-based model for learners to show the accounting system components and to simulate journal entries. The CPN model is composed of seven modules: configuration module, general ledger module, trial balance module, balance sheet module, and integrated module. All the CPN constructs of the accounting system modules correspond to a real accounting system, which has a configuring module of a chart of account and an account mapping profile. Learners can use the generated test cases to simulate the accounting processes:journal entry, ledger posting and report generation. The generated test cases or the CPN markings come from the application of the decision-table-based testing technique. There are 232 test cases for simulating the CPN model, generated from the the ground-truth data of a retail department store and mortgage loan system.

Before the use of the proposed CPN model in a class, we validated the CPN model with the generated test cases. Behaviors of the model are exercised to guarantee that it frees deadlock and corresponds to an actual accounting system. The experiments show that the CPN model can be used as a material for the instruction of accounting systems, accounting components, journal entries, and accounting transaction flows. The provided CPN model is an alternative way to help learners who are studying accounting systems. Because the simulating features of CPN tools may quite be difficult for beginners, our ongoing work is directed towards building the CPN model running on web browsers, in line with [27], in order to eliminate the drawbacks of the user interface and accessibility.

**Author Contributions:** Conceptualization, W.V. and C.D.; methodology, C.D.; software, C.D.; validation, C.D., W.V. and W.P.; formal analysis, C.D. and W.V.; investigation,C.D., W.V. and A.T.; resources, W.P.; data curation, C.D.; writing—original draft preparation, C.D.; writing—review and editing, C.D. and W.V.; visualization, C.D.; supervision, W.V. and A.T.; project administration, W.V.; funding acquisition, W.V. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Postdoctoral Fellowship (Ratchadaphiseksomphot Endowment Fund) of Chulalongkorn University.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The meta-model and test cases are private, but can be made available upon request.

**Acknowledgments:** The work of Chanon Dechsupa was supported by the Postdoctoral Fellowship(Ratchadaphiseksomphot Endowment Fund) of Chulalongkorn University. Thunyarat Thirawongpattana offered valuable data, and case studies which the authors used in this research.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Heiner, M.; Herajy, M.; Liu, F.; Rohr, C.; Schwarick, M. Snoopy—A unifying Petri net tool. In Proceedings of the International Conference on Application and Theory of Petri Nets and Concurrency, Hamburg, Germany, 25–29 June 2012; pp. 398–407.
2. Jensen, K.; Kristensen, L.M.; Wells, L. Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. *Int. J. Softw. Tools Technol. Transf.* **2007**, *9*, 213–254. [CrossRef]
3. Chiola, G.; Franceschinis, G.; Gaeta, R.; Ribaudo, M. GreatSPN 1.7: Graphical editor and analyzer for timed and stochastic Petri nets. *Perform. Eval.* **1995**, *24*, 47–68. [CrossRef]
4. Wolf, K. Petri net model checking with LoLA 2. In Proceedings of the International Conference on Applications and Theory of Petri Nets and Concurrency, Bratislava, Slovakia, 24–29 June2018; pp. 351–362.
5. Singh, H.; Conrad, M.; Sadeghipour, S. Test case design based on Z and the classification-tree method. In Proceedings of the First IEEE International Conference on Formal Engineering Methods, Hiroshima, Japan, 12–14 November1997; pp. 81–90.
6. Poolsawasdi, W.; Dechsupa, C. Formal Verification of the Accounting Information Interfaces Using Colored Petri Net. In Proceedings of the 2019 5th International Conference on Engineering, Applied Sciences and Technology (ICEAST), Luang Prabang, Laos, 2–5 July 2019; pp. 362–365.

7. Kim, R.; Gangolly, J.; Elsas, P. A framework for analytics and simulation of accounting information systems: A Petri net modeling primer. *Int. J. Account. Inf. Syst.* **2017**, *27*, 30–54. [CrossRef]
8. Yingwei, L.; Jiuzhi, M. The research of computerized accounting system of internal control. In Proceedings of the 2011 IEEE 3rd International Conference on Communication Software and Networks, Xi'an, China, 27–29 May 2011; pp. 302–304.
9. Szpyrka, M.; Szmuc, T. Decision tables in Petri net models. In Proceedings of the International Conference on Rough Sets and Intelligent Systems Paradigms, Warsaw, Poland, 28–30 June 2007; pp. 648–657.
10. Deesukying, J.; Vatanawood, W. Generating of business rules for Coloured Petri Nets. In Proceedings of the 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), Okayama, Japan, 26–29 June 2016; pp. 1–6.
11. Bredmar, K.; Ask, U.; Frisk, E.; Magnusson, J. Accounting Information Systems implementation and management accounting change. *Bus. Syst. Res. J.* **2014**, *5*, 125–138. [CrossRef]
12. Niu, Y. An empirical analysis of accounting information integration in integrated systems. In Proceedings of the 2010 2nd IEEE International Conference on Information Management and Engineering, Chengdu, China, 16–18 April 2010; pp. 107–110.
13. Zuama, R.A.; Hudin, J.M.; Puspitasari, D.; Hermaliani, E.H.; Riana, D. Quality dimensions of Delone-McLean model to measure students' accounting computer satisfaction: An empirical test on accounting system information. In Proceedings of the 2017 5th International Conference on Cyber and IT Service Management (CITSM), Denpasar, Indonesia, 8–10 August 2017; pp. 1–6.
14. Mauldin, E.G.; Ruchala, L.V. Towards a meta-theory of accounting information systems. *Organ. Soc.* **1999**, *24*, 317–331. [CrossRef]
15. Collins, D.; Deck, A.; McCrickard, M. Computer aided instruction A study of student evaluations and academic performance. *J. Coll. Teach. Learn.* **2008**, *5*, 49–58. [CrossRef]
16. Cooper, J.P.; Pattanayak, S. *Chart of Accounts: A Critical Element of the Public Financial Management Framework*; Citeseer: Princeton, NJ, USA, 2011.
17. Bryer, R.A. Double-entry bookkeeping and the birth of capitalism: Accounting for the commercial revolution in medieval northern Italy. *Crit. Perspect. Account.* **1993**, *4*, 113–140. [CrossRef]
18. Baier, C.; Katoen, J.P. *Principles of Model Checking*; MIT Press: Cambridge, MA, USA, 2008.
19. Fisher, M. *An Introduction to Practical Formal Methods Using Temporal Logic*; John Wiley & Sons: Hoboken, NJ, USA, 2011; Volume 82.
20. Clarke, E.M.; Klieber, W.; Nováček, M.; Zuliani, P. Model checking and the state explosion problem. In *LASER Summer School on Software Engineering*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 1–30.
21. Koch, I. Petri nets in systems biology. *Softw. Syst. Model.* **2015**, *14*, 703–710. [CrossRef]
22. Blätke, M.A.; Heiner, M.; Marwan, W. *Petri Nets in Systems Biology*; Technical Report; Otto von Guericke University Magdeburg: Magdeburg, Germany, 2011.
23. Pinney, J.W.; Westhead, D.R.; McConkey, G.A. Petri Net representations in systems biology. *Biochem. Soc. Trans.* **2003**, *31*, 1513–1515. [CrossRef] [PubMed]
24. Heiner Monika, D.G.; Donaldson, R. Petri nets for systems and synthetic biology. In Proceedings of the International School on Formal Methods for the Design of Computer, Communication and Software Systems, Bertinoro, Italy, 2–7 June 2008; pp. 215–264.
25. Liu, F. Colored Petri Nets for Systems Biology. Master's Thesis, Brandenburg University of Technology, Cottbus, Germany, 2012.
26. Jorgensen, P.C. *Software Testing: A Craftsman's Approach*; CRC Press: Boca Raton, FL, USA, 2018.
27. Computer Science Department, Boston University. SNOOPY: Web Animation. 2008. Available online: https://www-dssz.informatik.tu-cottbus.de/web_animation/pn/tram1.spped (accessed on 10 July 2021).