



**School of Computer Science & Engineering  
College of Design, Creative and Digital Industries  
University of Westminster**

## **6SENG006W – Concurrent Programming**

### **Coursework (2023/24)**

# **Printing System**

Lecturer: Tendai Mhlanga

Degree: BSc (Hons) in Software Engineering

Student: Shanmugaratnam Mohanaranjan  
(18705841/w1870584)

Date: 2024/01/12  
Time : 13.00

## FSP Main Program Code

```

File Edit Check Build Window Help Options
PRINTING_SYSTEM
Edit Output Draw Transitions Alphabet

// FSP Process Model to Ticket Machine, Passengers and Technicians.
// Shanmuganathan Mohanaranjan || w1870584 || 18705841.

// Declaring Constants
// Maximum capacity of sheets and toner in a printer.
// Maximum number of paper on a printer.
const MAXIMUM_NUMBER_OF_SHEETS = 3
// Maximum number of toner on a printer.
const MAXIMUM_NUMBER_OF_TONER = 3

// Number of sheets on a printer.
const NUMBER_OF_SHEETS = 0
// Number of toner on a printer.
const NUMBER_OF_TONER = 0

// Maximum number of sheets on a printer.
const MAXIMUM_OF_AMOUNT = 1
// Maximum number of sheets on a printer.
const MAXIMUM_OF_AMOUNT = 3

// Declaring Ranges
// Range representing the number of sheets in the printer.
range SHEETS_OF_RANGE = NUMBER_OF_SHEETS .. MAXIMUM_NUMBER_OF_SHEETS
// Range representing the number of toner cartridges in the printer.
range RANGE_OF_TONER = NUMBER_OF_TONER .. MAXIMUM_NUMBER_OF_TONER
// Range representing the number of prints.
range PRINT_AMOUNT_RANGE = MINIMUM_OF_AMOUNT .. MAXIMUM_OF_AMOUNT

// Relevant Sets
// Set of actions including acquiring prints, printing, acquiring, releasing, and passing counts for sheets and toner.
set PRINT_ACTIONS = (acquirePrint[SHEETS_OF_RANGE][RANGE_OF_TONER], print, acquireRefill, refill, release, passPaperCount[SHEETS_OF_RANGE], passTonerCount[RANGE_OF_TONER])
// Set of users including passengers and technicians.
set USERS = {passenger1, passenger2, passenger3, passenger4, toner_technician, paper_technician}

// Ticket Machine Process
// Represents the ticket machine process. It can acquire prints, print, acquire/refill, release, and pass counts for sheets and toner.
// It has conditions to check if there's enough paper or toner, and actions accordingly.
PRINTER[COUNT_OF_PAPER = MAXIMUM_NUMBER_OF_SHEETS, TONER_COUNT = MAXIMUM_NUMBER_OF_TONER] =
    PRINTER[paper: NUMBER_OF_SHEETS..COUNT_OF_PAPER][toner: NUMBER_OF_TONER..TONER_COUNT] =
        // Condition to check the paper and toner
        if(paper == NUMBER_OF_SHEETS || toner == NUMBER_OF_TONER)
            then(acquireRefill --> refill --> release --> PRINTER[paper][MAXIMUM_NUMBER_OF_SHEETS][toner])
            | acquireRefill --> refill --> release --> PRINTER[paper][MAXIMUM_NUMBER_OF_TONER]
        else(acquirePrint[paper][toner] --> print --> release --> passPaperCount[paper - 1] --> passTonerCount[toner - 1] --> PRINTER[paper - 1][toner - 1])

// Paper Technician Process
// Represents the paper technician process.
// It can pass paper counts and either initiate a refill process for paper or wait.
// It includes common print actions.
PAPER_TECHNICIAN =
    (passPaperCount[paper: SHEETS_OF_RANGE] --> if(paper == NUMBER_OF_SHEETS)
        // Refill Process for Paper
        then(acquireRefill --> refill --> release --> PAPER_TECHNICIAN)
        else(wait --> PAPER_TECHNICIAN)) + PRINT_ACTIONS.

// Toner Technician Process
// Represents the toner technician process.
// It can pass toner counts and either initiate a refill process for toner or wait.
// It includes common print actions.
TONER_TECHNICIAN =
    (passTonerCount[toner: RANGE_OF_TONER] --> if(toner == NUMBER_OF_TONER)
        // Refill Process for Toner
        then(acquireRefill --> refill --> release --> TONER_TECHNICIAN)
        else(wait --> TONER_TECHNICIAN)) + PRINT_ACTIONS.

// Passenger Process
// Represents the passenger process.
// It can pass print demands and either initiate a print process or wait.
// It includes common print actions.
PASSENGER[TICKETS = 1] =
    PASSENGER[demand: PRINT_AMOUNT_RANGE] =
        (acquirePrint[paper: SHEETS_OF_RANGE][toner: RANGE_OF_TONER] --> if(toner >= demand && paper >= demand)
            then(print --> release --> PASSENGER)
            else(wait --> PASSENGER)) + PRINT_ACTIONS.

// Printing System Process
// Represents the overall printing system with mutual exclusive control of the ticket machine.
// It involves multiple passengers, paper/toner technicians, and printer processes.
// Mutual exclusive control of the Ticket Machine
|| PRINTING_SYSTEM [
    || passenger1 : PASSENGER(2)
    || passenger2 : PASSENGER(1)
    || passenger3 : PASSENGER(3)
    || passenger4 : PASSENGER(1)
    || paper_technician : PAPER_TECHNICIAN
    || toner_technician : TONER_TECHNICIAN
    || USERS :: PRINTER].

```

```

File Edit Check Build Window Help Options
PRINTING_SYSTEM
Edit Output Draw Transitions Alphabet

// Ticket Machine Process
// Represents the printer process. It can acquire prints, print, acquire/refill, release, and pass counts for sheets and toner.
// It has conditions to check if there's enough paper or toner, and actions accordingly.
PRINTER[COUNT_OF_PAPER = MAXIMUM_NUMBER_OF_SHEETS, TONER_COUNT = MAXIMUM_NUMBER_OF_TONER] =
    PRINTER[paper: NUMBER_OF_SHEETS..COUNT_OF_PAPER][toner: NUMBER_OF_TONER..TONER_COUNT] =
        // Condition to check the paper and toner
        if(paper == NUMBER_OF_SHEETS || toner == NUMBER_OF_TONER)
            then(acquireRefill --> refill --> release --> PRINTER[MAXIMUM_NUMBER_OF_SHEETS][toner])
            | acquireRefill --> refill --> release --> PRINTER[paper][MAXIMUM_NUMBER_OF_TONER]
        else(acquirePrint[paper][toner] --> print --> release --> passPaperCount[paper - 1] --> passTonerCount[toner - 1] --> PRINTER[paper - 1][toner - 1]).

// Paper Technician Process
// Represents the paper technician process.
// It can pass paper counts and either initiate a refill process for paper or wait.
// It includes common print actions.
PAPER_TECHNICIAN =
    (passPaperCount[paper: SHEETS_OF_RANGE] --> if(paper == NUMBER_OF_SHEETS)
        // Refill Process for Paper
        then(acquireRefill --> refill --> release --> PAPER_TECHNICIAN)
        else(wait --> PAPER_TECHNICIAN)) + PRINT_ACTIONS.

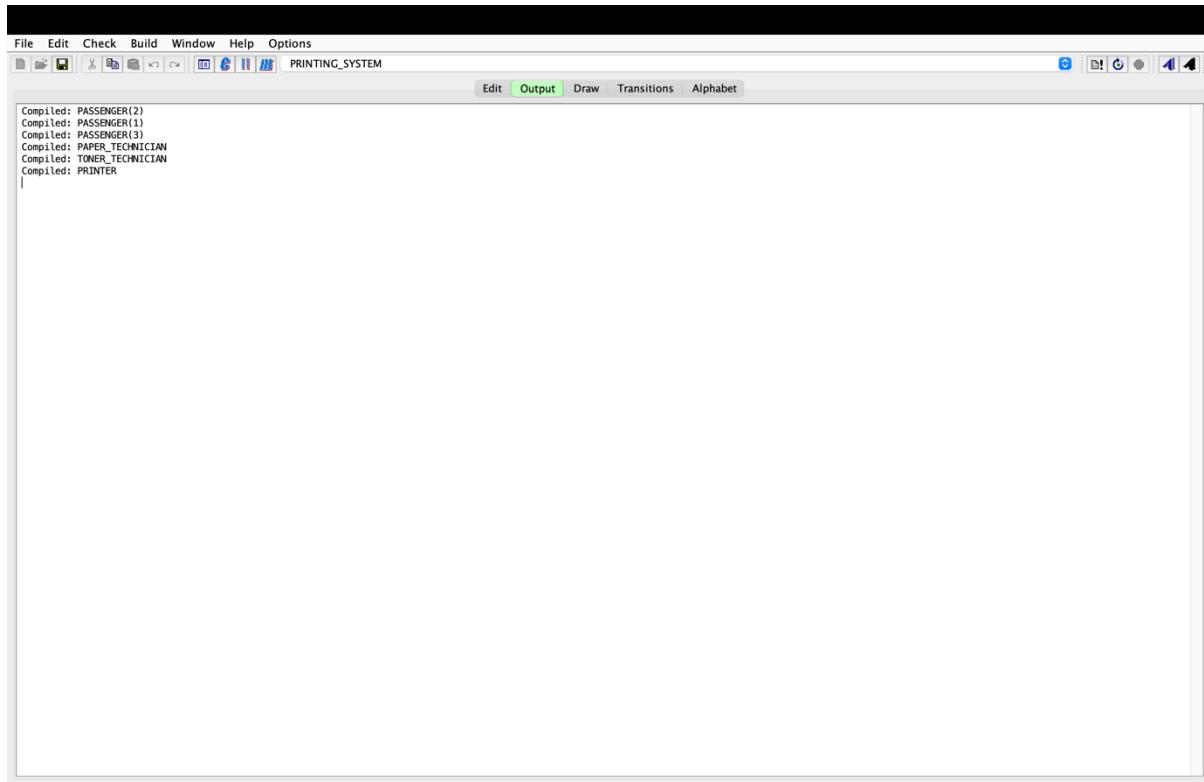
// Toner Technician Process
// Represents the toner technician process.
// It can pass toner counts and either initiate a refill process for toner or wait.
// It includes common print actions.
TONER_TECHNICIAN =
    (passTonerCount[toner: RANGE_OF_TONER] --> if(toner == NUMBER_OF_TONER)
        // Refill Process for Toner
        then(acquireRefill --> refill --> release --> TONER_TECHNICIAN)
        else(wait --> TONER_TECHNICIAN)) + PRINT_ACTIONS.

// Passenger Process
// Represents the passenger process.
// It can pass print demands and either initiate a print process or wait.
// It includes common print actions.
PASSENGER[TICKETS = 1] =
    PASSENGER[demand: PRINT_AMOUNT_RANGE] =
        (acquirePrint[paper: SHEETS_OF_RANGE][toner: RANGE_OF_TONER] --> if(toner >= demand && paper >= demand)
            then(print --> release --> PASSENGER)
            else(wait --> PASSENGER)) + PRINT_ACTIONS.

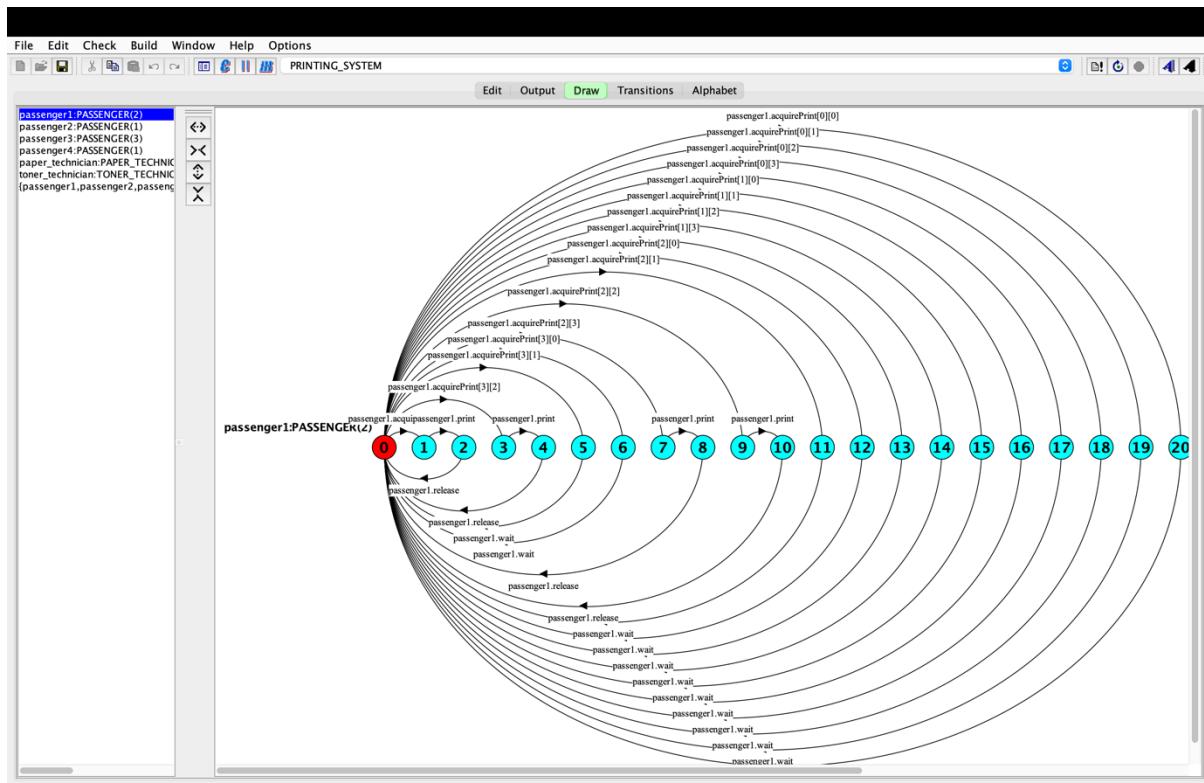
// Printing System Process
// Represents the overall printing system with mutual exclusive control of the ticket machine.
// It involves multiple passengers, paper/toner technicians, and printer processes.
// Mutual exclusive control of the Ticket Machine
|| PRINTING_SYSTEM [
    || passenger1 : PASSENGER(2)
    || passenger2 : PASSENGER(1)
    || passenger3 : PASSENGER(3)
    || passenger4 : PASSENGER(1)
    || paper_technician : PAPER_TECHNICIAN
    || toner_technician : TONER_TECHNICIAN
    || USERS :: PRINTER].

```

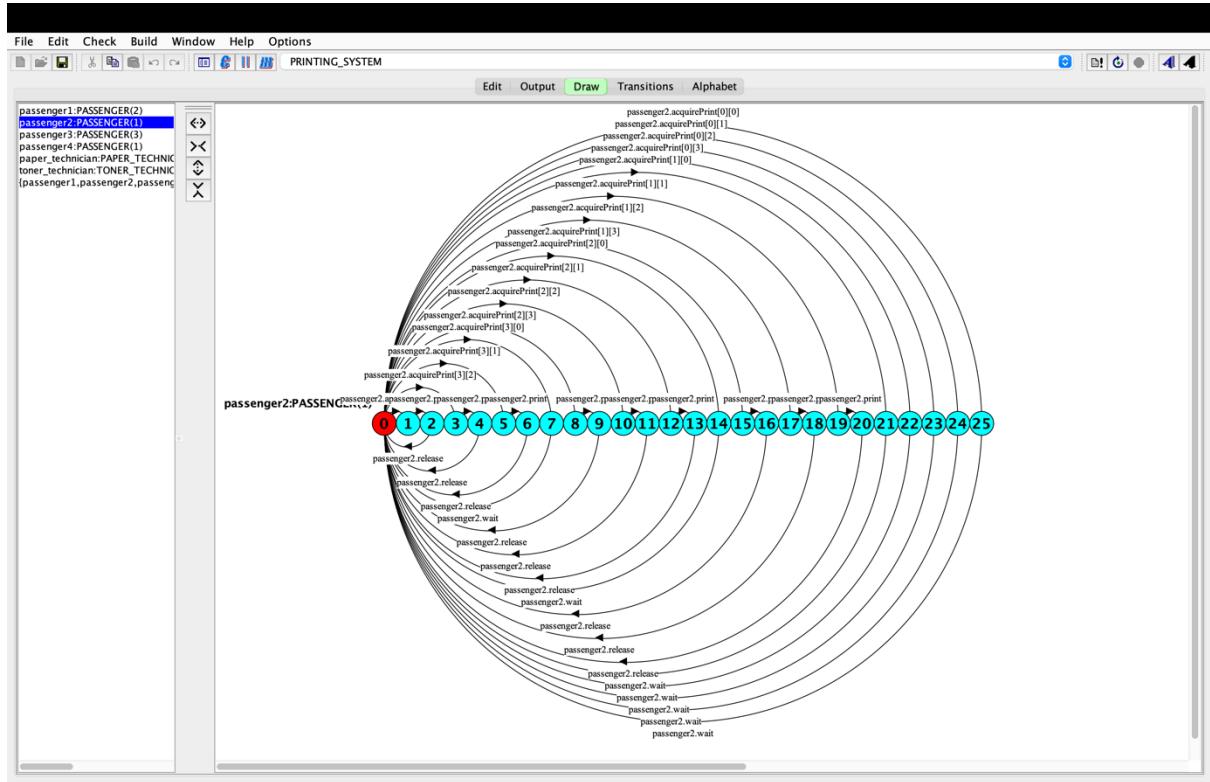
## Compile



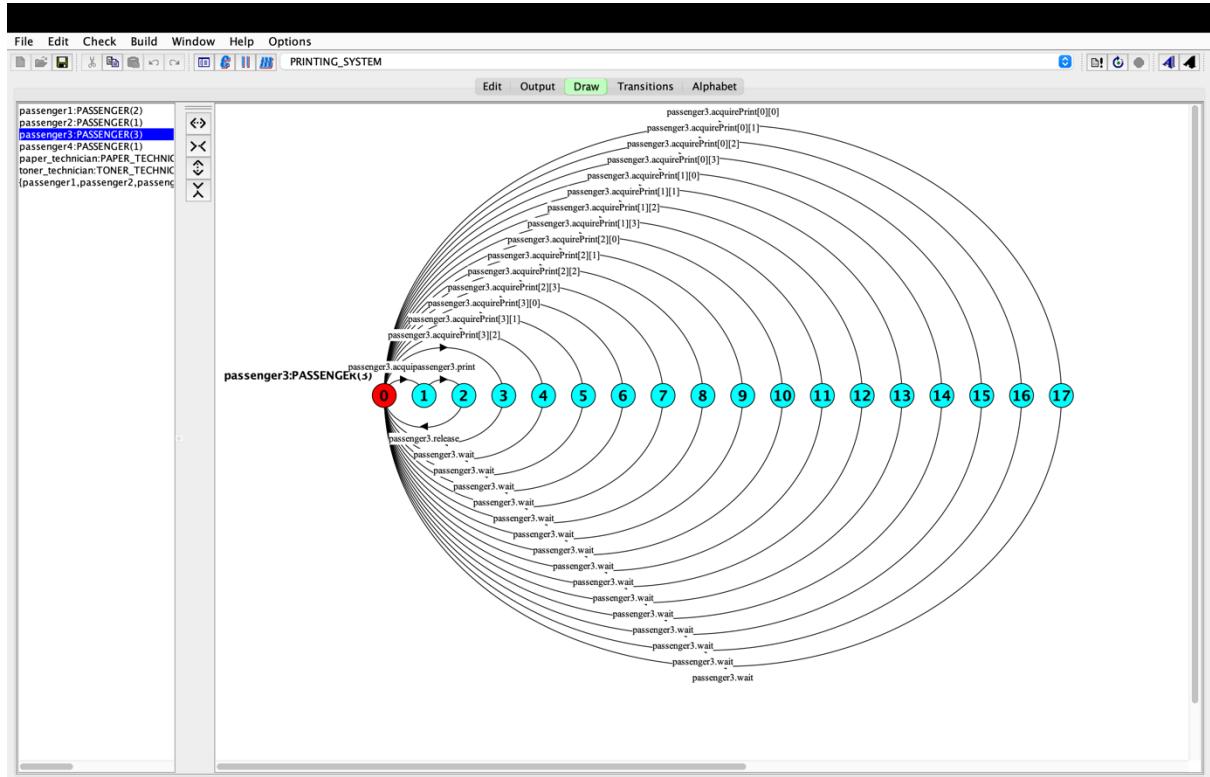
## Passenger 1



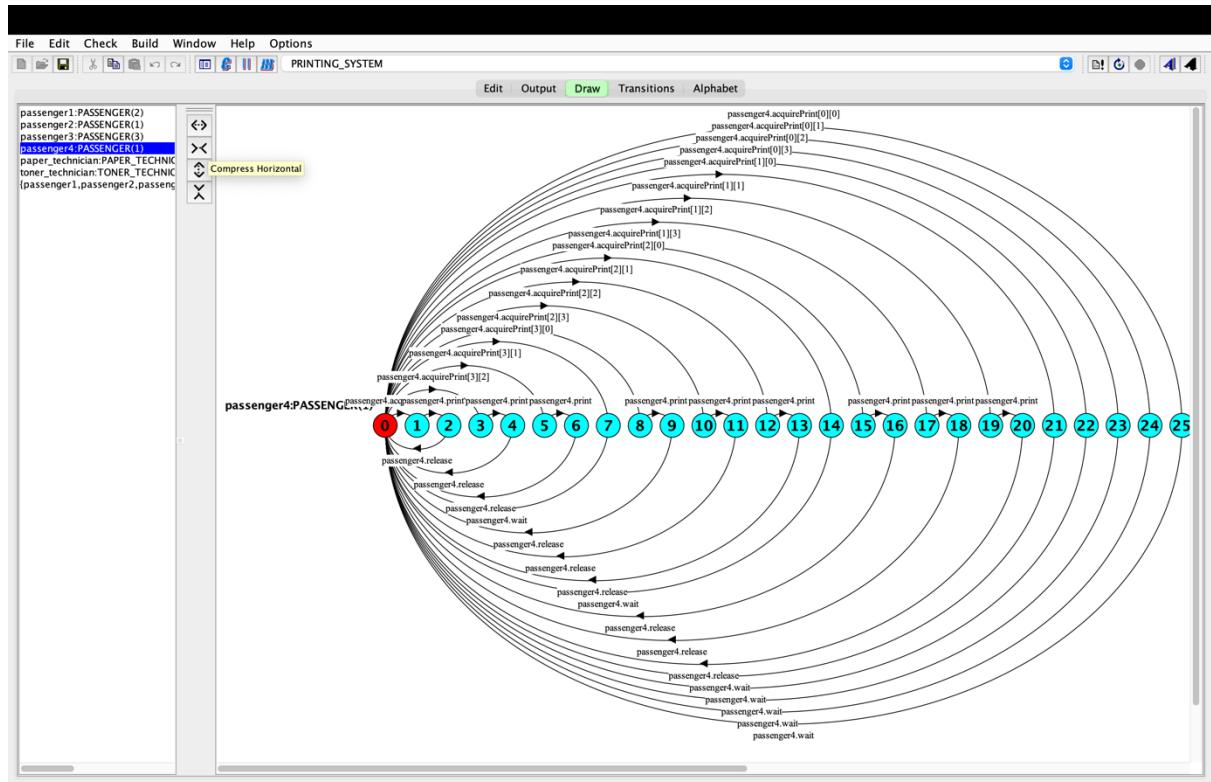
## Passenger 2



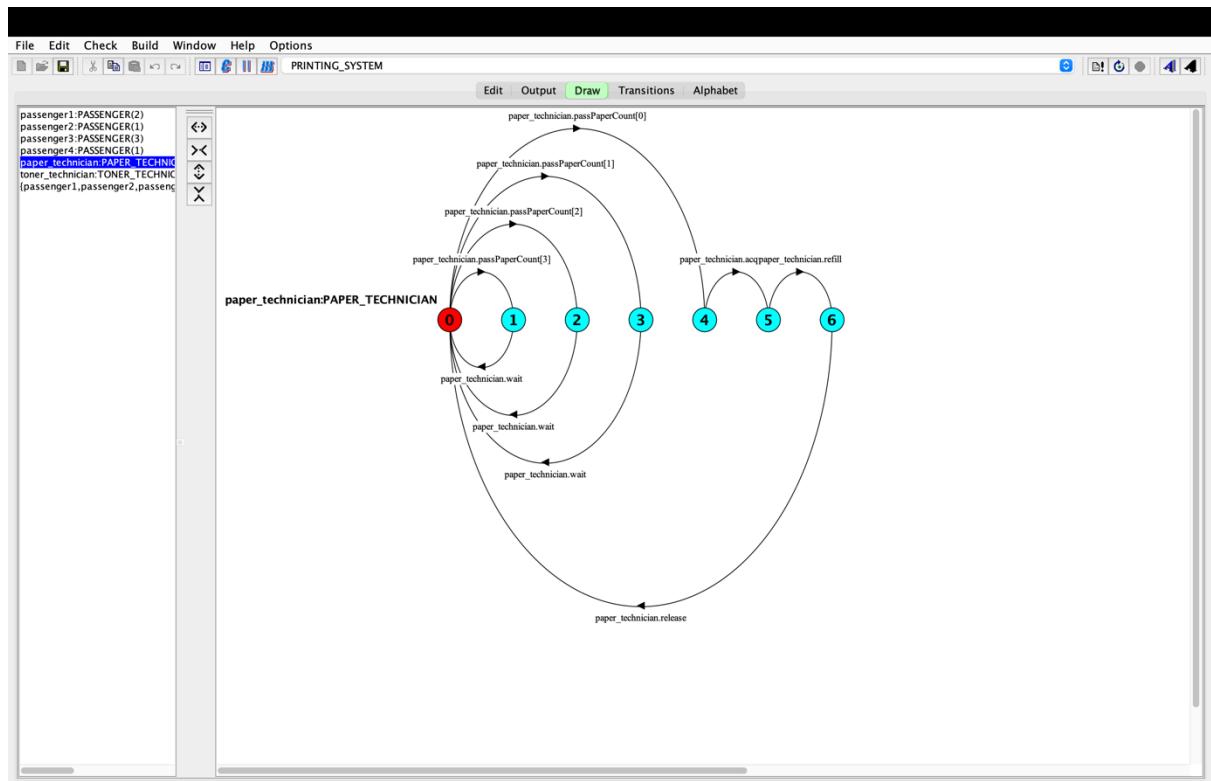
## Passenger 3



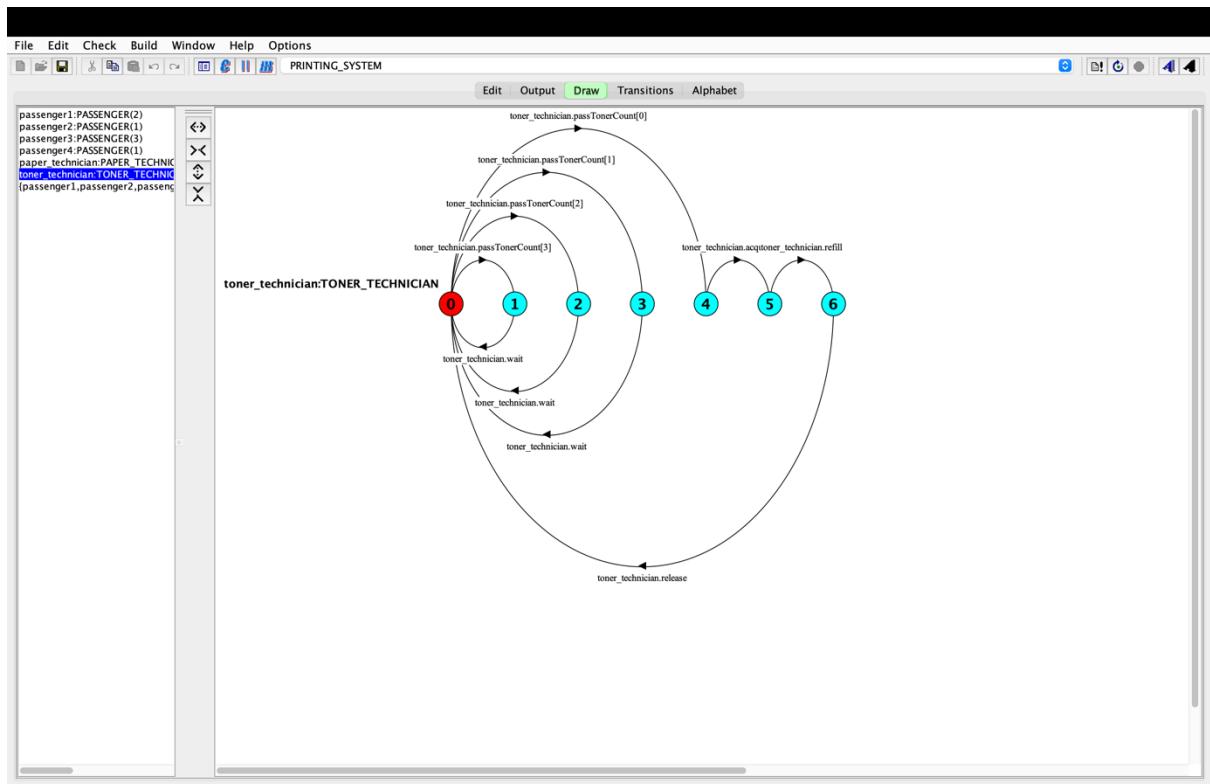
Passenger 4



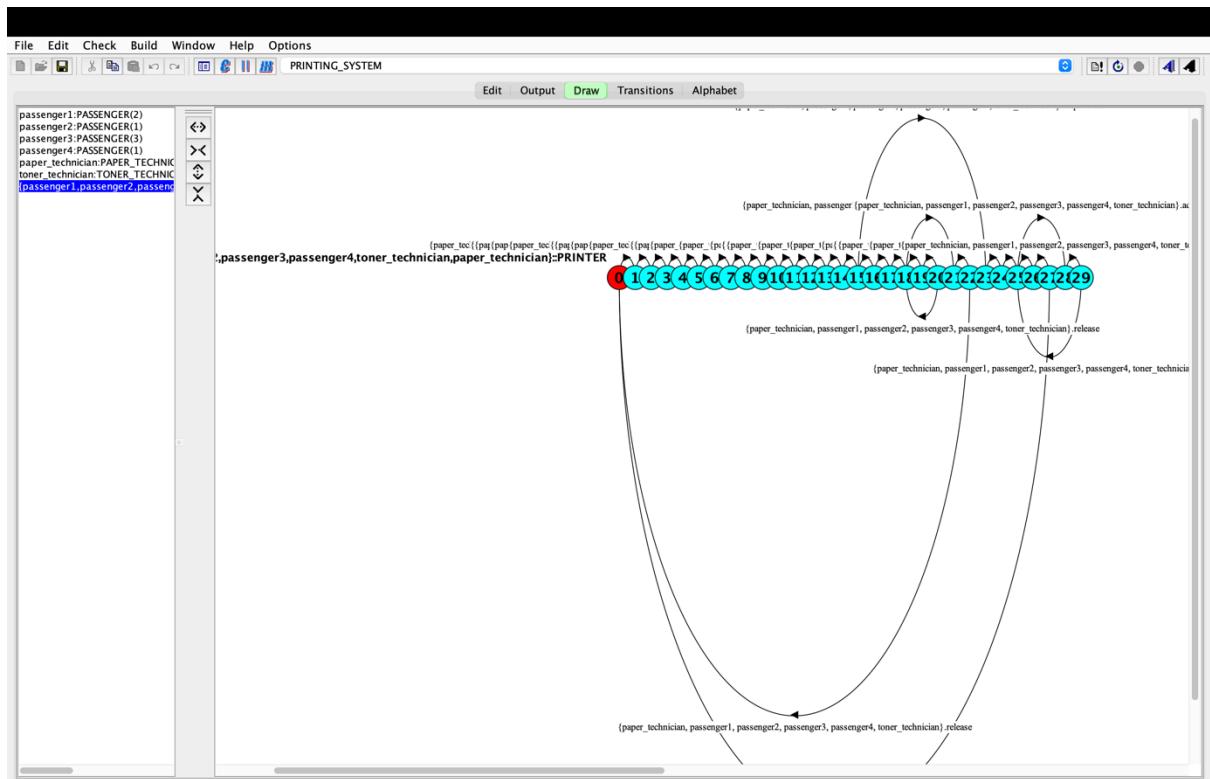
## Paper Technician



## Toner Technician



Printer



## Alphabet Passenger 1

```
File Edit Check Build Window Help Options
PRINTING_SYSTEM
Edit Output Draw Transitions Alphabet
Process:
passenger1:PASSENGER()
passenger2:PASSENGER()
passenger3:PASSENGER()
passenger4:PASSENGER()
paper_technician:PAPER_
toner_technician:TONER_
{passenger1,passenger2}
Al
+
-
1
Alphabets:
passenger1.(acquirePrint[0..3], acquireRefill, {passPaperCount, passTonerCount}[0..3], {print, refill, release, wait})
```

## Passenger 2

```
File Edit Check Build Window Help Options
PRINTING_SYSTEM
Edit Output Draw Transitions Alphabet
Process:
passenger2:PASSENGER(1)
Alphabets:
passenger2.(acquirePrint[0..3], acquireRefill, {passPaperCount, passTonerCount}[0..3], {print, refill, release, wait})
Al
+
-
1
```

### Passenger 3

The screenshot shows a software interface with a menu bar (File, Edit, Check, Build, Window, Help, Options) and a toolbar with various icons. The main window title is "PRINTING\_SYSTEM". The tab "Alphabet" is selected. On the left, there is a tree view with nodes: passenger1:PASSENGER(), passenger2:PASSENGER(), passenger3:PASSENGER(), passenger4:PASSENGER(), paper\_technician:PAPER(), toner\_technician:TONER\_, and {passenger1,passenger2}. A context menu is open over the passenger3 node, showing options: All, +, -, 1. The right panel displays the process definition:

```
Process:  
  passenger3:PASSENGER(3)  
Alphabet:  
  passenger3.{acquirePrint[0..3][0..3], acquireRefill, {passPaperCount, passTonerCount}[0..3], {print, refill, release, wait}}
```

### Passenger 4

The screenshot shows a software interface with a menu bar (File, Edit, Check, Build, Window, Help, Options) and a toolbar with various icons. The main window title is "PRINTING\_SYSTEM". The tab "Alphabet" is selected. On the left, there is a tree view with nodes: passenger1:PASSENGER(), passenger2:PASSENGER(), passenger3:PASSENGER(), passenger4:PASSENGER(), paper\_technician:PAPER(), toner\_technician:TONER\_, and {passenger1,passenger2}. A context menu is open over the passenger4 node, showing options: All, +, -, 1. The right panel displays the process definition:

```
Process:  
  passenger4:PASSENGER(1)  
Alphabet:  
  passenger4.{acquirePrint[0..3][0..3], acquireRefill, {passPaperCount, passTonerCount}[0..3], {print, refill, release, wait}}
```

## Paper Technician

The screenshot shows a software interface with a menu bar (File, Edit, Check, Build, Window, Help, Options) and a toolbar with various icons. The main window title is "PRINTING\_SYSTEM". The tab "Alphabet" is selected. On the left, there is a tree view with nodes: passenger1:PASSENGER(), passenger2:PASSENGER(), passenger3:PASSENGER(), passenger4:PASSENGER(), paper\_technician:PAPER(), toner\_technician:TONER(), (passenger1,passenger2). A panel on the right displays the process definition:

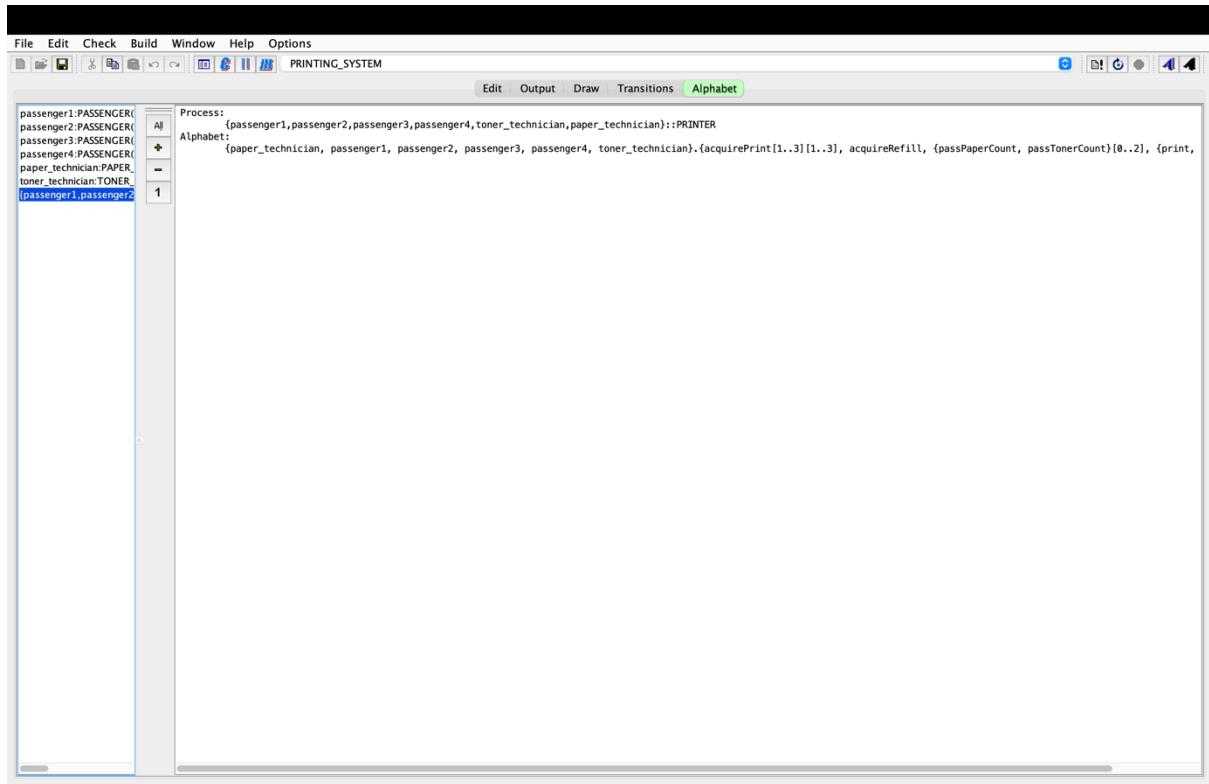
```
Process: paper_technician:PAPER_TECHNICIAN
Alphabet:
paper_technician.{acquirePrint[0..3][0..3], acquireRefill, {passPaperCount, passTonerCount}{0..3}, {print, refill, release, wait}}
```

## Toner Technician

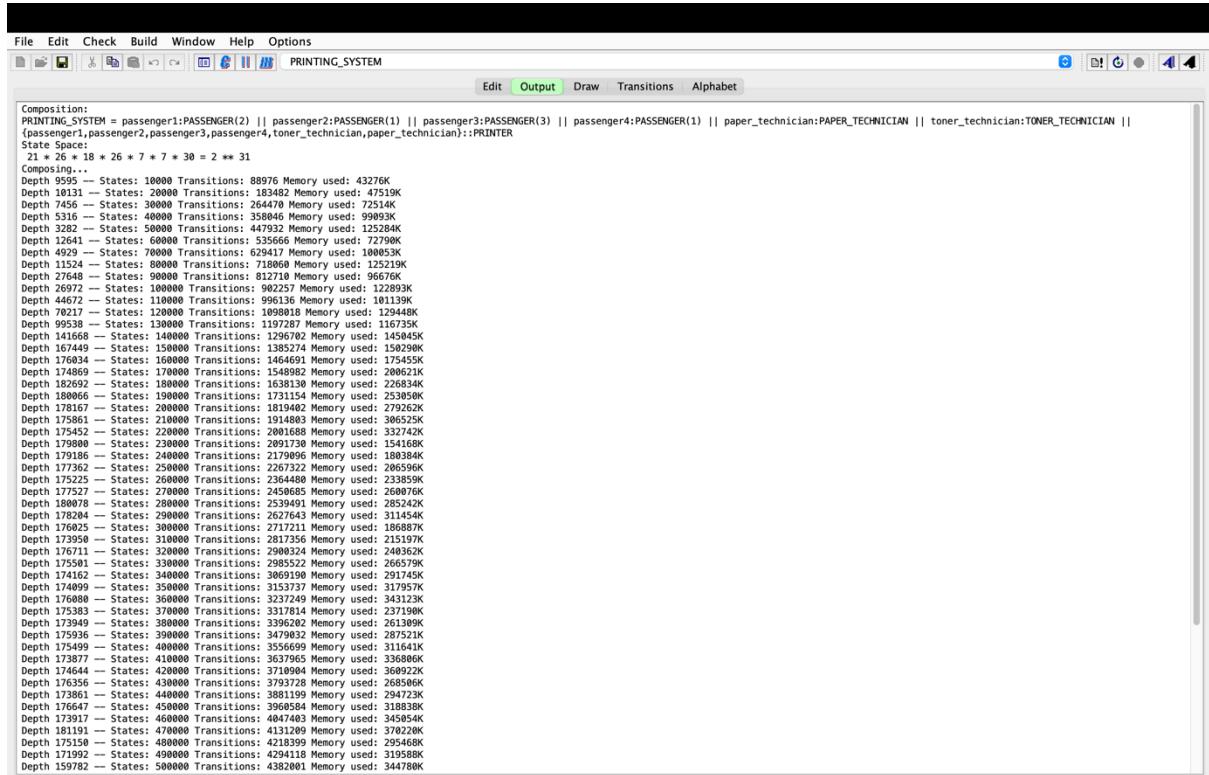
The screenshot shows a software interface with a menu bar (File, Edit, Check, Build, Window, Help, Options) and a toolbar with various icons. The main window title is "PRINTING\_SYSTEM". The tab "Alphabet" is selected. On the left, there is a tree view with nodes: passenger1:PASSENGER(), passenger2:PASSENGER(), passenger3:PASSENGER(), passenger4:PASSENGER(), paper\_technician:PAPER(), toner\_technician:TONER(), (passenger1,passenger2). A panel on the right displays the process definition:

```
Process: toner_technician:TONER_TECHNICIAN
Alphabet:
toner_technician.{acquirePrint[0..3][0..3], acquireRefill, {passPaperCount, passTonerCount}{0..3}, {print, refill, release, wait}}
```

## Printer



## State Space



## Number of States

```

File Edit Check Build Window Help Options
PRINTING_SYSTEM
Edit Output Draw Transitions Alphabet
Depth 44672 — States: 110000 Transitions: 996136 Memory used: 101139K
Depth 76217 — States: 120000 Transitions: 1098018 Memory used: 129448K
Depth 107652 — States: 130000 Transitions: 1272703 Memory used: 135338K
Depth 141668 — States: 140000 Transitions: 1352570 Memory used: 145528K
Depth 167449 — States: 150000 Transitions: 1385274 Memory used: 159299K
Depth 176034 — States: 160000 Transitions: 1464691 Memory used: 17455K
Depth 178869 — States: 170000 Transitions: 1548982 Memory used: 20621K
Depth 182692 — States: 180000 Transitions: 1638138 Memory used: 226834K
Depth 186566 — States: 190000 Transitions: 1731150 Memory used: 253058K
Depth 189181 — States: 200000 Transitions: 1840440 Memory used: 279528K
Depth 175061 — States: 210000 Transitions: 1914803 Memory used: 305528K
Depth 175452 — States: 220000 Transitions: 2001688 Memory used: 332742K
Depth 179898 — States: 230000 Transitions: 2091730 Memory used: 154168K
Depth 179186 — States: 240000 Transitions: 2179896 Memory used: 188384K
Depth 177362 — States: 250000 Transitions: 2267322 Memory used: 206596K
Depth 175225 — States: 260000 Transitions: 2364488 Memory used: 233859K
Depth 174099 — States: 270000 Transitions: 2461654 Memory used: 251987K
Depth 180078 — States: 280000 Transitions: 2539491 Memory used: 285242K
Depth 178284 — States: 290000 Transitions: 2627643 Memory used: 311454K
Depth 176025 — States: 300000 Transitions: 2717211 Memory used: 186887K
Depth 173925 — States: 310000 Transitions: 2817356 Memory used: 215197K
Depth 176711 — States: 320000 Transitions: 2900324 Memory used: 248362K
Depth 175781 — States: 330000 Transitions: 2985340 Memory used: 266029K
Depth 174152 — States: 340000 Transitions: 3061508 Memory used: 297745K
Depth 174099 — States: 350000 Transitions: 3152737 Memory used: 317957K
Depth 176088 — States: 360000 Transitions: 3237249 Memory used: 343123K
Depth 175384 — States: 370000 Transitions: 3317814 Memory used: 237190K
Depth 173949 — States: 380000 Transitions: 3396202 Memory used: 261308K
Depth 175924 — States: 390000 Transitions: 3479832 Memory used: 287521K
Depth 176191 — States: 400000 Transitions: 3563561 Memory used: 313144K
Depth 173877 — States: 410000 Transitions: 3637965 Memory used: 336806K
Depth 174644 — States: 420000 Transitions: 3718984 Memory used: 368922K
Depth 176356 — States: 430000 Transitions: 3793728 Memory used: 268506K
Depth 173861 — States: 440000 Transitions: 3881199 Memory used: 294723K
Depth 176647 — States: 450000 Transitions: 3968584 Memory used: 318838K
Depth 175191 — States: 460000 Transitions: 4047600 Memory used: 345922K
Depth 174191 — States: 470000 Transitions: 4126209 Memory used: 372228K
Depth 175150 — States: 480000 Transitions: 4218399 Memory used: 205468K
Depth 171992 — States: 490000 Transitions: 4294118 Memory used: 319588K
Depth 159782 — States: 500000 Transitions: 4382801 Memory used: 344780K
Depth 148088 — States: 510000 Transitions: 4473609 Memory used: 372016K
Depth 136062 — States: 520000 Transitions: 4567869 Memory used: 398229K
Depth 129173 — States: 530000 Transitions: 4658000 Memory used: 425522K
Depth 127208 — States: 540000 Transitions: 4755768 Memory used: 457758K
Depth 100587 — States: 550000 Transitions: 4851816 Memory used: 489018K
Depth 88835 — States: 560000 Transitions: 4954165 Memory used: 507281K
Depth 77173 — States: 570000 Transitions: 5055591 Memory used: 535594K
Depth 65498 — States: 580000 Transitions: 5158917 Memory used: 562857K
Depth 53894 — States: 590000 Transitions: 5262187 Memory used: 591167K
Depth 422859 — States: 600000 Transitions: 5374457 Memory used: 619494K
Depth 40859 — States: 610000 Transitions: 5469229 Memory used: 640449K
Depth 19462 — States: 620000 Transitions: 5575463 Memory used: 675853K
Depth 8154 — States: 630000 Transitions: 5684354 Memory used: 704413K
— States: 637448 Transitions: 5782144 Memory used: 620527K
Composed in 3034ms

```

## No Deadlocks

```

File Edit Check Build Window Help Options
PRINTING_SYSTEM
Edit Output Draw Transitions Alphabet
No deadlocks/errors

```

## No Deadlock Trace

```

File Edit Check Build Window Help Options
PRINTING_SYSTEM
Edit Output Draw Transitions Alphabet
Depth 12 -- States: 120000 Transitions: 1159083 Memory used: 206715K
Depth 12 -- States: 130000 Transitions: 1250548 Memory used: 65307K
Depth 13 -- States: 140000 Transitions: 1352003 Memory used: 120409K
Depth 13 -- States: 150000 Transitions: 1428928 Memory used: 112491K
Depth 13 -- States: 160000 Transitions: 1519851 Memory used: 135559K
Depth 14 -- States: 170000 Transitions: 1611868 Memory used: 158628K
Depth 14 -- States: 180000 Transitions: 1693557 Memory used: 180650K
Depth 14 -- States: 190000 Transitions: 1786554 Memory used: 204765K
Depth 15 -- States: 200000 Transitions: 1877077 Memory used: 227834K
Depth 15 -- States: 210000 Transitions: 1968600 Memory used: 250856K
Depth 16 -- States: 220000 Transitions: 2055606 Memory used: 273971K
Depth 16 -- States: 230000 Transitions: 2141996 Memory used: 297040K
Depth 17 -- States: 240000 Transitions: 2244322 Memory used: 322296K
Depth 17 -- States: 250000 Transitions: 2329635 Memory used: 345275K
Depth 18 -- States: 260000 Transitions: 2433758 Memory used: 369394K
Depth 18 -- States: 270000 Transitions: 2522893 Memory used: 392462K
Depth 19 -- States: 280000 Transitions: 2614021 Memory used: 415579K
Depth 19 -- States: 290000 Transitions: 2713797 Memory used: 440607K
Depth 19 -- States: 300000 Transitions: 2795968 Memory used: 462715K
Depth 19 -- States: 310000 Transitions: 2894396 Memory used: 486834K
Depth 20 -- States: 320000 Transitions: 2988324 Memory used: 510950K
Depth 20 -- States: 330000 Transitions: 3063449 Memory used: 532972K
Depth 21 -- States: 340000 Transitions: 3140574 Memory used: 554997K
Depth 21 -- States: 350000 Transitions: 3225937 Memory used: 674677K
Depth 21 -- States: 360000 Transitions: 3326496 Memory used: 89488K
Depth 22 -- States: 370000 Transitions: 3419522 Memory used: 112554K
Depth 22 -- States: 380000 Transitions: 3496159 Memory used: 134576K
Depth 23 -- States: 390000 Transitions: 3585968 Memory used: 157644K
Depth 23 -- States: 400000 Transitions: 3667281 Memory used: 180713K
Depth 24 -- States: 410000 Transitions: 3760404 Memory used: 203806K
Depth 24 -- States: 420000 Transitions: 3852043 Memory used: 227097K
Depth 25 -- States: 430000 Transitions: 3927107 Memory used: 250966K
Depth 26 -- States: 440000 Transitions: 4027619 Memory used: 275085K
Depth 26 -- States: 450000 Transitions: 4126752 Memory used: 300251K
Depth 27 -- States: 460000 Transitions: 4246843 Memory used: 327514K
Depth 27 -- States: 470000 Transitions: 4359757 Memory used: 352680K
Depth 28 -- States: 480000 Transitions: 4473700 Memory used: 377812K
Depth 28 -- States: 490000 Transitions: 4583392 Memory used: 404058K
Depth 28 -- States: 500000 Transitions: 4668124 Memory used: 427126K
Depth 28 -- States: 510000 Transitions: 4757528 Memory used: 451246K
Depth 28 -- States: 520000 Transitions: 4842224 Memory used: 475361K
Depth 28 -- States: 530000 Transitions: 4932784 Memory used: 498430K
Depth 29 -- States: 540000 Transitions: 5042324 Memory used: 522464K
Depth 29 -- States: 550000 Transitions: 5152862 Memory used: 545188K
Depth 29 -- States: 560000 Transitions: 5265038 Memory used: 68438K
Depth 29 -- States: 570000 Transitions: 5267626 Memory used: 107401K
Depth 29 -- States: 580000 Transitions: 5343426 Memory used: 138478K
Depth 29 -- States: 590000 Transitions: 5438422 Memory used: 153539K
Depth 30 -- States: 600000 Transitions: 5510194 Memory used: 175561K
Depth 30 -- States: 610000 Transitions: 5601364 Memory used: 179514K
Depth 30 -- States: 620000 Transitions: 5658544 Memory used: 16698K
Depth 31 -- States: 630000 Transitions: 5730874 Memory used: 119720K
Depth 32 -- States: 637440 Transitions: 5782144 Memory used: 135446K
No deadlocks/errors
Analysed in: 944ms

```

## Java Code

```

import utils.Utilities;
import utils.Utilities.ProcessLogger.*;
public class PrintingSystem {
    public static void main(String[] args) {
        //create Thread Groups
        Utilities.printLogs(PRINTING_SYSTEM, message: "Creating Thread Groups ", Utilities.ProcessLogger.INFO)
        ThreadGroup passengerGroup = new ThreadGroup( name: "Passenger");
        Utilities.printLogs(PRINTING_SYSTEM, message: " Passenger Thread Group Created", Utilities.ProcessLog
        ThreadGroup technicianGroup = new ThreadGroup( name: "Technician");
        Utilities.printLogs(PRINTING_SYSTEM, message: " Technician Thread Group Created", Utilities.ProcessLog
        //create PrinterMachine Resource
        PrinterTicketMachine printer = new PrinterTicketMachine( printerID: "PrinterMachine 1120J", passengerGrou
        Utilities.printLogs(PRINTING_SYSTEM, message: "Initialised PRINTER Resource". Utilities.ProcessLogger.
    }
}

```

Output Log:

```

[2024-01-12 07:59:28.325] [PRINTER] [INFO] Printer{printerID='PrinterMachine 1120J', paper_level=247, toner_level=447, printedTickets=4}.
[2024-01-12 07:59:28.326] [PRINTER] [INFO] Paper level has not reached the minimum level to be refilled. Waiting to check again in 5 seconds.
[2024-01-12 07:59:28.362] [PASSENGER] [INFO] Pasan generated Ticket to print: Ticket[ UserID: Pasan, Name: tick_id_2, Prints: 6].
[2024-01-12 07:59:28.362] [PASSENGER] [INFO] [Pasan] has requested to print the ticket: Ticket[ UserID: Pasan, Name: tick_id_2, Prints: 6].
[2024-01-12 07:59:28.363] [PRINTER] [INFO] Printer{printerID='PrinterMachine 1120J', paper_level=247, toner_level=447, printedTickets=4}.
[2024-01-12 07:59:28.363] [PRINTER] [INFO] Printed Document : Ticket[ UserID: Pasan, Name: tick_id_2, Prints: 6].
[2024-01-12 07:59:28.363] [PRINTER] [INFO] Printer{printerID='PrinterMachine 1120J', paper_level=241, toner_level=441, printedTickets=5}.
[2024-01-12 07:59:28.363] [PASSENGER] [INFO] Printer current status: Printer{printerID='PrinterMachine 1120J', paper_level=241, toner_level=441, printedTickets=5}.
[2024-01-12 07:59:28.363] [PRINTER] [INFO] Printer{printerID='PrinterMachine 1120J', paper_level=241, toner_level=441, printedTickets=5}.
[2024-01-12 07:59:28.364] [PRINTER] [INFO] Printer{printerID='PrinterMachine 1120J', paper_level=241, toner_level=441, printedTickets=5}.

```