6SENG002W Concurrent Programming

FSP Process Composition Analysis & Design Form

Name	Pasan Sandaru Kottearachchi
Student ID	W1761885
Date	12/22/2022

1. FSP Composition Process Attributes

Attribute	**Value**
Name	PRINTING_SYSTEM
Description	Models the process of printer, two students and one technician. One student(stud_1) trying to print three documents, another student(stud_2) trying to print two document and technician(tech) refilling the sheets as required.
Alphabet (Use LTSA's compressed notation, if alphabet is large.)	{{stud_1, stud_2}.{documentPrint[1..3], stud.{acquire, release}, technician.{acquire, refillPrinterSheets, release}}, tech.{checkAvailableSheets, documentPrint[1..3], stud.{acquire, release}, technician.{acquire, refillPrinterSheets, release}}}
Sub-processes (List them.)	STUDENT, PRINTER, TECHNICIAN
Number of States	55
Deadlocks (yes/no)	No
Deadlock Trace(s) (If applicable)	N/A

2. FSP "main" Program Code

The code for the parallel composition of all of the sub-processes and the definitions of any constants, ranges & process labelling sets used. (Do not include the code for the other sub-processes.)

FSP Program:

```
//Define the CONSTANTS of the system
const MAX_NO_OF_PAPER = 3
range PAPER_RANGE = 1 .. MAX_NO_OF_PAPER

 //Set of actions
set USERS = {stud_1, stud_2, tech }
set Print_Actions = {stud.acquire, documentPrint[PAPER_RANGE], stud.release,
technician.refillPrinterSheets, technician.acquire, technician.release}

 //Print COMPOSITE Finite State Machines
|| PRINTING_SYSTEM = (stud_1:STUDENT(3) || stud_2:STUDENT(2) || tech: TECHNICIAN ||
{USERS}::PRINTER) .
```

3. Combined Sub-processes

(Add rows as necessary.)

Process	Description
PRINTER	Models the printer can hold three sheets for print.
STUDENT(3)	Models the student trying to print three documents.
STUDENT(2)	Models the student trying to print two documents.
TECHNICIAN	Technician refill the printer when the sheets as required.

4. Analysis of Combined Process Actions

- **Synchronous** actions are performed by at least two sub-process in the combination.
- **Blocked Synchronous** actions cannot be performed, since at least one of the sub-processes cannot preform them, because they were added to their alphabet using alphabet extension.
- **Asynchronous** actions are preformed independently by a single sub-process.

Group actions together if appropriate, for example if they include indexes,
e.g. in[0], in[1], …, in[5] as in[1..5].

(Add rows as necessary.)

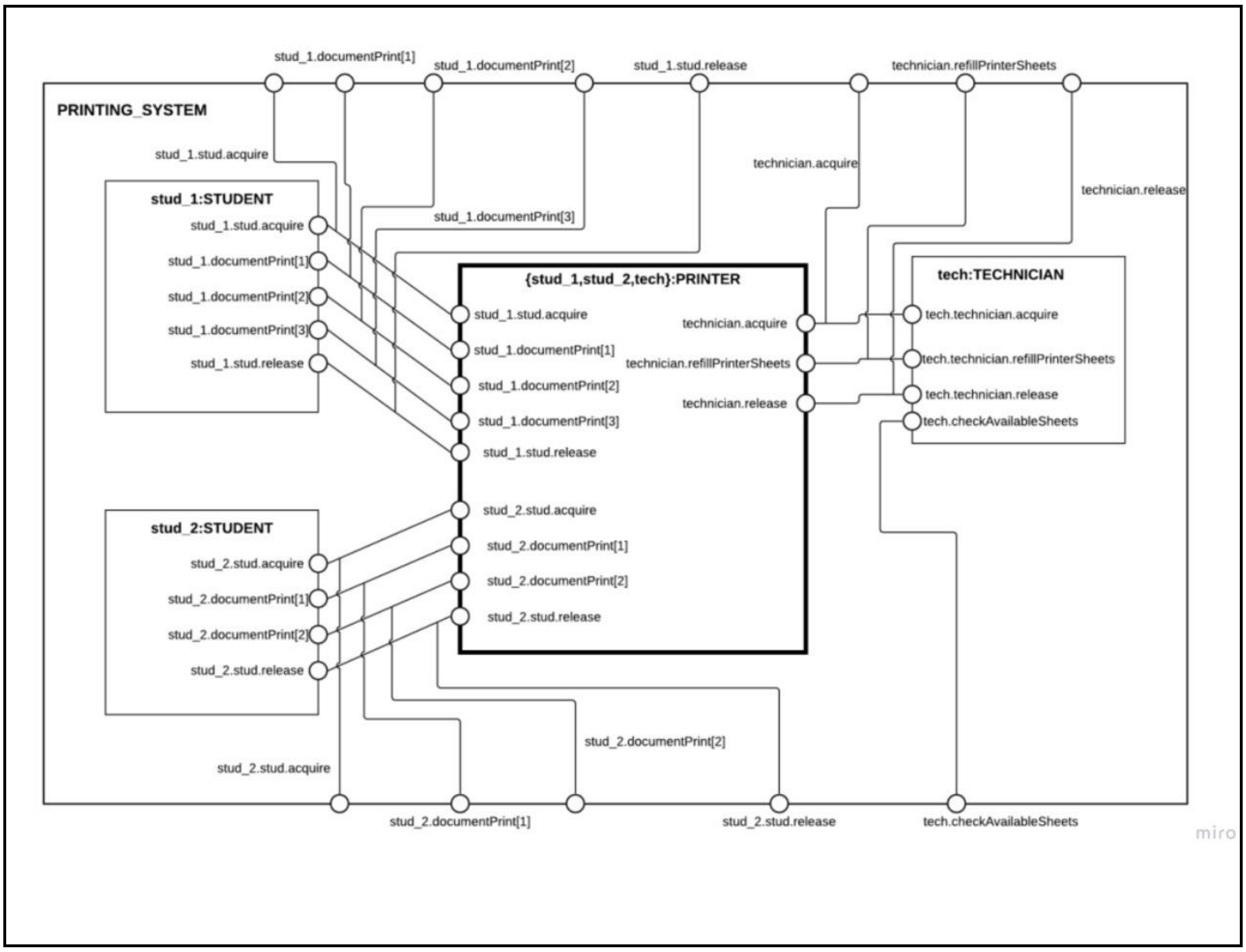
Synchronous Actions	Synchronised by Sub-Processes (List)
stud_1.stud.acquire, stud_1.documentPrint[1], stud_1.documentPrint[2], stud_1.documentPrint[3], stud_1.stud.release	STUDENT(3), PRINTER
stud_2.stud.acquire, stud_2.documentPrint[1], stud_2.documentPrint[2], stud_2.stud.release	STUDENT(2), PRINTER
tech.technician.acquire, tech.technician.refillPrinterSheets tech.technician.release	TECHNICIAN, PRINTER

Block Synchronous Actions	Synchronised by Sub-Processes (List)	Blocking Sub-Processes
stud_1.stud.acquire, stud_1.documentPrint[1..3], stud_1.stud.release	STUDENT(3), PRINTER	TECHNICIAN, STUDENT(2)
stud_2.stud.acquire, stud_2.documentPrint[1..2], stud_2.stud.release	STUDENT(2), PRINTER	TECHNICIAN, STUDENT(3)
tech.technician.acquire, tech.technician.refillPrinterSheets tech.technician.release	TECHNICIAN, PRINTER	STUDENT(3), STUDENT(2)

Sub-Process	Asynchronous Actions (List)
TECHNICIAN	tech.checkAvailableSheets

5. Parallel Composition Structure Diagram

The structure diagram for the parallel composition.


PRINTING_SYSTEM
stud_1.documentPrint[1]
stud_1.documentPrint[2]
stud_1.stud.release
technician.refillPrinterSheets
stud_1.stud.acquire
technician.acquire
technician.release
stud_1.documentPrint[3]
stud_1:STUDENT
stud_1.stud.acquire
stud_1.documentPrint[1]
stud_1.documentPrint[2]
stud_1.documentPrint[3]
stud_1.stud.release
{stud_1,stud_2,tech}:PRINTER
stud_1.stud.acquire
stud_1.documentPrint[1]
stud_1.documentPrint[2]
stud_1.documentPrint[3]
stud_1.stud.release
stud_2.stud.acquire
stud_2.documentPrint[1]
stud_2.documentPrint[2]
stud_2.stud.release
technician.acquire
technician.refillPrinterSheets
technician.release
tech:TECHNICIAN
tech.technician.acquire
tech.technician.refillPrinterSheets
tech.technician.release
tech.checkAvailableSheets
stud_2:STUDENT
stud_2.stud.acquire
stud_2.documentPrint[1]
stud_2.documentPrint[2]
stud_2.stud.release
stud_2.documentPrint[2]
stud_2.stud.acquire
stud_2.documentPrint[1]
stud_2.stud.release
tech.checkAvailableSheets
miro