6SENG002W Concurrent Programming

FSP Process Composition Analysis & Design Form

Name	M.G. Vajith Chamuditha
Student ID	2019437
Date	12/01/2023

1. FSP Composition Process Attributes

Attribute	Value
Name	PRINTING_SYSTEM
Description	This process involves merging two different STUDENT processes, one TECHNICIAN process, and one PRINTER process all running simultaneously. A set of process prefix labels and relabelling of the composite process is employed to guarantee exclusive access to the printer.
Alphabet (Use LTSA's compressed notation, if alphabet is large.)	{std1.print, std2.print , std1.printLock, std2.printLock, std1.refill, std2.refill, std1.refillLock, std2.refillLock, std1.release, std2.release, tech.print, tech.printLock, tech.refill, tech.refillLock, tech.release, tech.wait}

Sub-processes
(List them.)
	std1:STUDENT(3)
	std2:STUDENT(2)
	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:

```
const MAX_SHEETS = 3
range SHEET_RANGE =0..MAX_SHEETS
set PRINTER_ACTIONS = {printLock, print, refillLock, refill, release}
set Users = {std1 , std2, tech}

||PRINTING_SYSTEM = (std1: STUDENT(3) || std2: STUDENT(2) || tech: TECHNICIAN ||
Users:: PRINTER).
```

3. Combined Sub-processes

(Add rows as necessary.)

Process	Description
std1:STUDENT(3)	An instance of the STUDENT process called 'std1' is used with a value of 3 as its parameter. The name 'std1' is given to this process to ensure that it has a unique identifier and does not share an alphabet with another instance of the STUDENT process named 'std2'.
Std2:STUDENT(2)	An instance of the STUDENT process called 'std2' is used with a value of 2 as its parameter. The name 'std2' is given to this process to ensure that it has a unique identifier and does not share an alphabet with another instance of the STUDENT process named 'std12'.
TECHNICIAN	TECHNICIAN process.
PRINTER	The PRINTER process has a label prefix of USERS, this allows the PRINTER's actions to be shared, this way the actions of the PRINTER can be synchronized separately with both the STUDENT and Technician processes.

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].

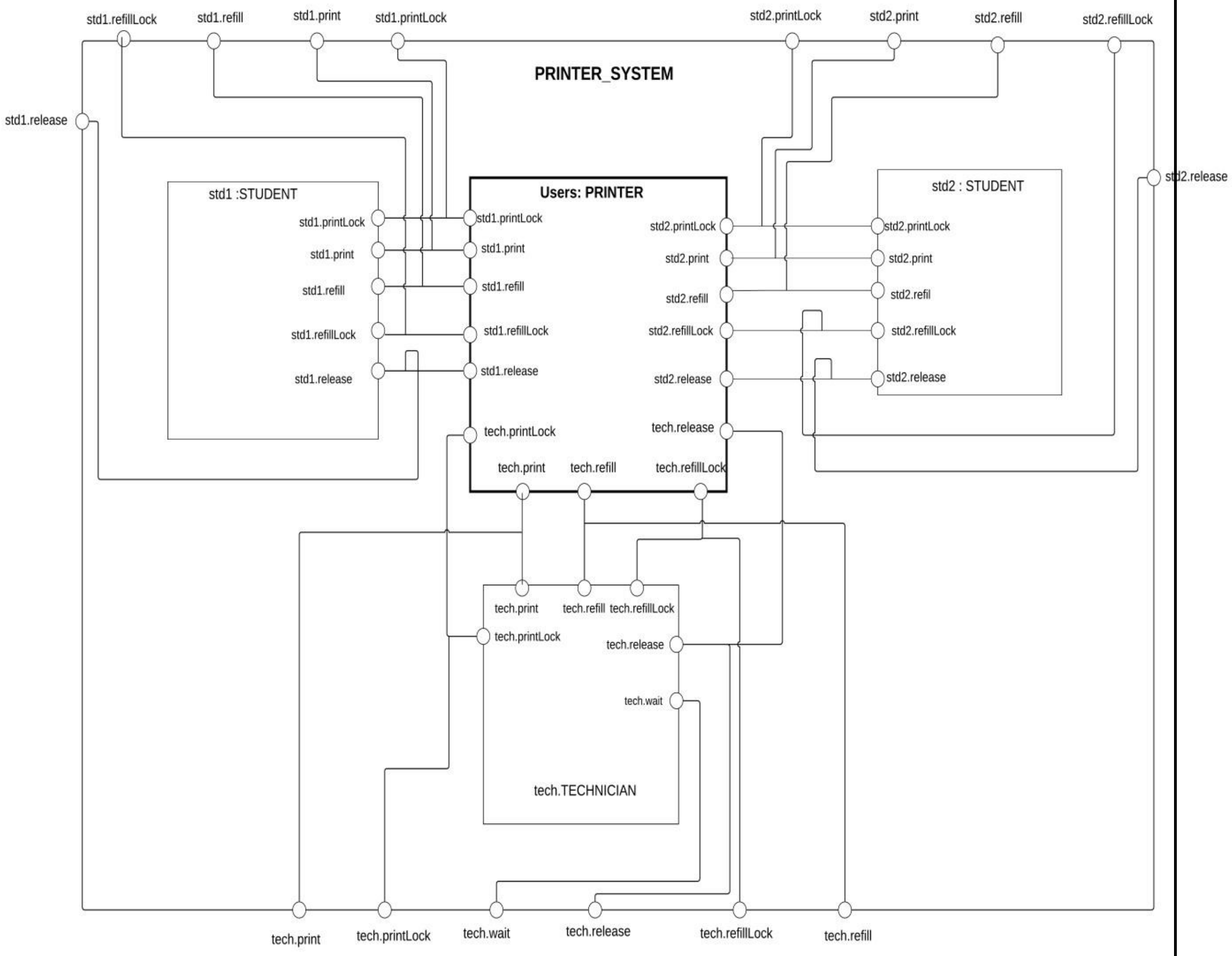
(Add rows as necessary.)

Synchronous Actions	Synchronised by Sub-Processes (List)
std1.{print , refill, printLock, release }	std1: STUDENT(3), PRINTER
std2.{print , refill, printLock, release }	std2: STUDENT(2), PRINTER
technician.{ print, refill, printLock, refillLock, release }	TECHNICIAN, PRINTER
terminate	std1: STUDENT(3), std2: STUDENT(2), TECHNICAIN

Sub-Process	Asynchronous Actions (List)
std1: STUDENT(3)	std1. print [1..3]
std2: STUDENT(2)	std2. print [1..2]
PRINTER	None
TECHNICIAN	None

5. Parallel Composition Structure Diagram

The structure diagram for the parallel composition.


std1.refillLock
std1.refill
std1.print
std1.printLock
std2.printLock
std2.print
std2.refill
std2.refillLock
PRINTER_SYSTEM
std1.release
std2.release
std1 :STUDENT
std1.printLock
std1.print
std1.refill
std1.refillLock
std1.release
Users: PRINTER
std1.printLock
std1.print
std1.refill
std1.refillLock
std1.release
tech.printLock
std2.printLock
std2.print
std2.refill
std2.refillLock
std2.release
tech.release
tech.print
tech.refill
tech.refillLock
std2 : STUDENT
std2.printLock
std2.print
std2.refil
std2.refillLock
std2.release
tech.print
tech.refill
tech.refillLock
tech.printLock
tech.release
tech.wait
tech.TECHNICIAN
tech.print
tech.printLock
tech.wait
tech.release
tech.refillLock
tech.refill