6SENG002W Concurrent Programming

FSP Process Composition Analysis & Design Form

Name	Tharindu De Silva
Student ID	W1761890/2018367
Date	January 2023

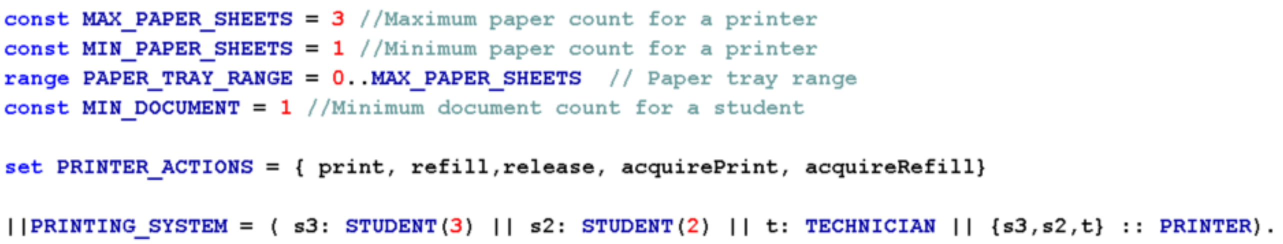
1. FSP Composition Process Attributes

Attribute	**Value**
Name	PRINTING_SYSTEM
Description	This is a composite process model of a printing system. It includes two student process and a technician process.
Alphabet (Use LTSA's compressed notation, if alphabet is large.)	{{s2, s3}.{acquirePrint, acquireRefill, print, refill, release}, t.{acquirePrint, acquireRefill, print, refill, release, wait}}
Sub-processes (List them.)	PRINTER, s1: STUDENT, s2: STUDENT, TECHNICIAN
Number of States	55
Deadlocks (yes/no)	No
Deadlock Trace(s) **(If applicable)**	Not applicable

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_PAPER_SHEETS = 3 //Maximum paper count for a printer
const MIN_PAPER_SHEETS = 1 //Minimum paper count for a printer
range PAPER_TRAY_RANGE = 0..MAX_PAPER_SHEETS  // Paper tray range
const MIN_DOCUMENT = 1 //Minimum document count for a student

set PRINTER_ACTIONS = { print, refill,release, acquirePrint, acquireRefill}

||PRINTING_SYSTEM = ( s3: STUDENT(3) || s2: STUDENT(2) || t: TECHNICIAN || {s3,s2,t} :: PRINTER).
```

3. Combined Sub-processes

(Add rows as necessary.)

Process	Description
s1: STUDENT	This sub-process is to model the behaviour of a student who uses the printer to print documents. This instance of a student process is wished to print 3 documents.
s2: STUDENT	This sub-process is to model the behaviour of a student who uses the printer to print documents. This instance of a student process is wished to print 3 documents.
TECHNICIAN	This sub-process models the behaviour of a technician who refills the printer with paper when it needs to be refilled.
PRINTER	This sub process models the behaviour of the printer.

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 perform 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.)

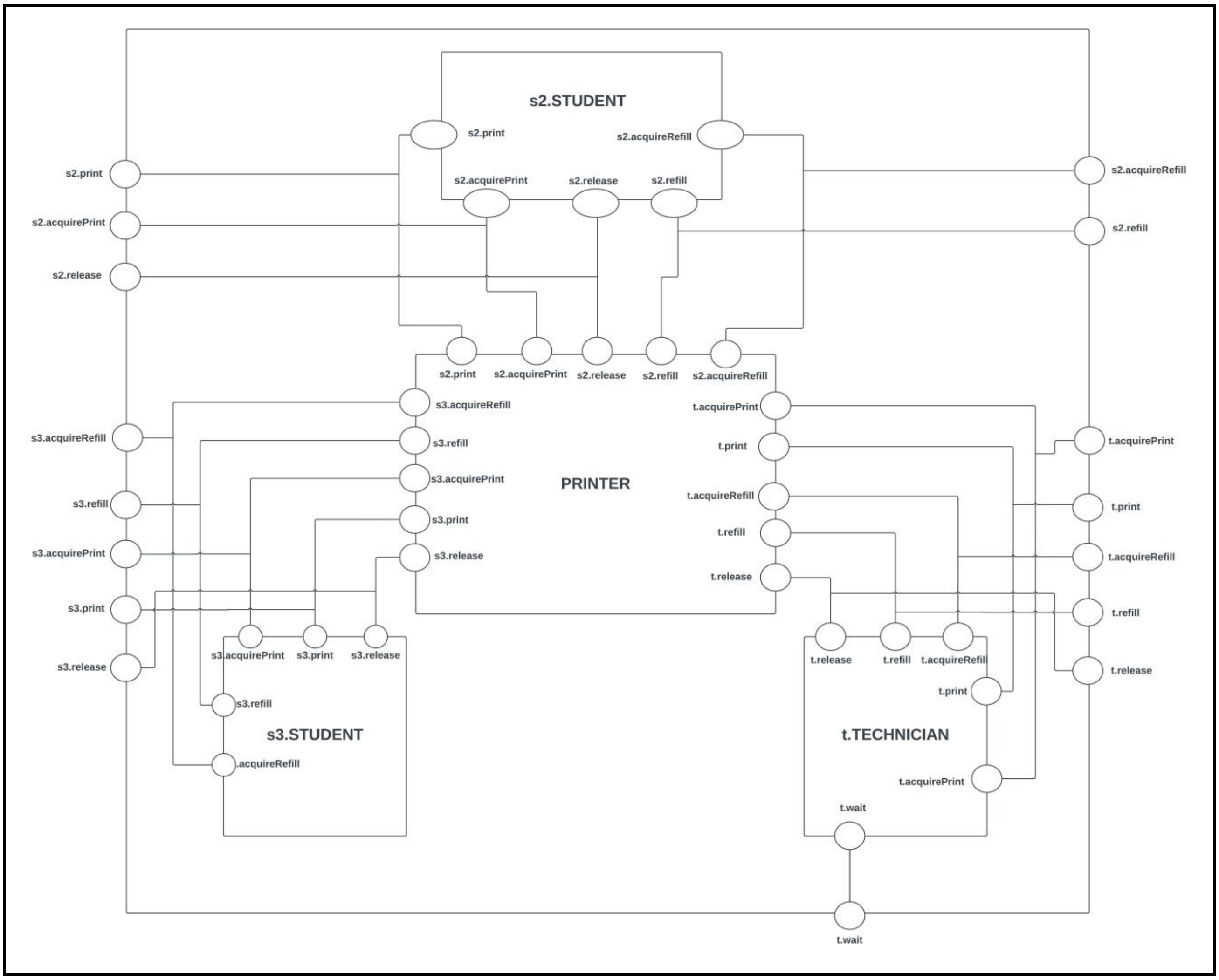
Synchronous Actions	**Synchronised by Sub-Processes (List)**
s2.acquirePrint, s2.print	S2:STUDENT(2), PRINTER
s3.acquirePrint, s3.print	S3:STUDENT(3), PRINTER
s2.release, t.release	t:TECHNICIAN, S3:STUDENT(3), PRINTER
S3.release, t.release	t:TECHNICIAN, S3:STUDENT(2), PRINTER
t.acquireRefil, t.refill	t:TECHNICIAN, PRINTER

Blocked Synchronous Actions	**Synchronizing Sub process (List)**	**Blocking sub-process**
s2.acuireRefill, s2.refill	S2:STUDENT(2), PRINTER	S2:STUDENT(2)
s3.acquireRefil, s3.refill	S3:STUDENT(3), PRINTER	S3:STUDENT(3)
t.acquirePrint, t.print	t.TECHNICIAN, PRINTER	t.TECHNICIAN

Sub-Process	**Asynchronous Actions (List)**
S2:STUDENT(2)	Not applicable
S3:STUDENT(3)	Not applicable
PRINTER	Not applicable
t.TECHNICIAN	t.wait

5. Parallel Composition Structure Diagram

The structure diagram for the parallel composition.


s2.STUDENT
s2.print
s2.acquireRefill
s2.acquirePrint
s2.release
s2.refill
s2.print
s2.acquirePrint
s2.release
s2.acquireRefill
s2.refill
s2.print
s2.acquirePrint
s2.release
s2.refill
s2.acquireRefill
PRINTER
s3.acquireRefill
s3.refill
s3.acquirePrint
s3.print
s3.release
t.acquirePrint
t.print
t.acquireRefill
t.refill
t.release
s3.acquireRefill
s3.refill
s3.acquirePrint
s3.print
s3.release
t.acquirePrint
t.print
t.acquireRefill
t.refill
t.release
s3.acquirePrint
s3.print
s3.release
s3.refill
s3.STUDENT
.acquireRefill
t.release
t.refill
t.acquireRefill
t.print
t.TECHNICIAN
t.acquirePrint
t.wait
t.wait