6SENG002W Concurrent Programming

FSP Process Composition Analysis & Design Form

Name	Warsha Vimanga Kiringoda
Student ID	UoW- W1697817 IIT- 2017366
Date	02/01/2021

1. FSP Composition Process Attributes

Attribute	Value
Name	PRINTING_SYSTEM
Description	This is the composite process which combines 2 different STUDENT process, 1 TECHNICIAN process and 1 PRINTER process in parallel. A process prefix label set, and composite process relabelling is used to ensure the mutually exclusive access to the printer.
Alphabet (Use LTSA's compressed notation, if alphabet is large.)	alphabet(PRINTING_SYSTEM) = {{a,b}.{print,stuAccess,stuRelease},techAccess,techRelease}

Sub-processes
(List them.)
	a:STUDENT(3)
	b:STUDENT(2)
	TECHNICIAN
	PRINTER

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_PAPERS = 3
range PAPER_COUNT = 0..MAX_PAPERS
const DEFAULT_NUM_OF_DOCS = 0

||PRINTING_SYSTEM = (a:STUDENT(3) || b:STUDENT(2) || TECHNICIAN ||
{a,b}::PRINTER)/{techAccess/{a,b}.techAccess,techRelease/{a,b}.techRelease}.
```

3. Combined Sub-processes

(Add rows as necessary.)

Process	Description
a:STUDENT(3)	An instance of STUDENT process named as '**a'**, that takes 3 as the value of its parameter. The process has the prefix '**a**', to make sure this process has a disjoint alphabet from the other b:STUDENT process.
b:STUDENT(2)	An instance of STUDENT process named as '**b**', that takes 2 as the value of its parameter. The process has the prefix '**b'**, to make sure this process has a disjoint alphabet from the other a:STUDENT process.
TECHNICIAN	TECHNICIAN process.
{a,b}::PRINTER	PRINTER process with process prefix label set {a,b}. This is done to make the printer's actions **shared actions**, so that the PRINTER's actions synchronize with each STUDENT process separately. (A relabelling is then done to secure PRINTER's synchronization with TECHNICIAN).

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].

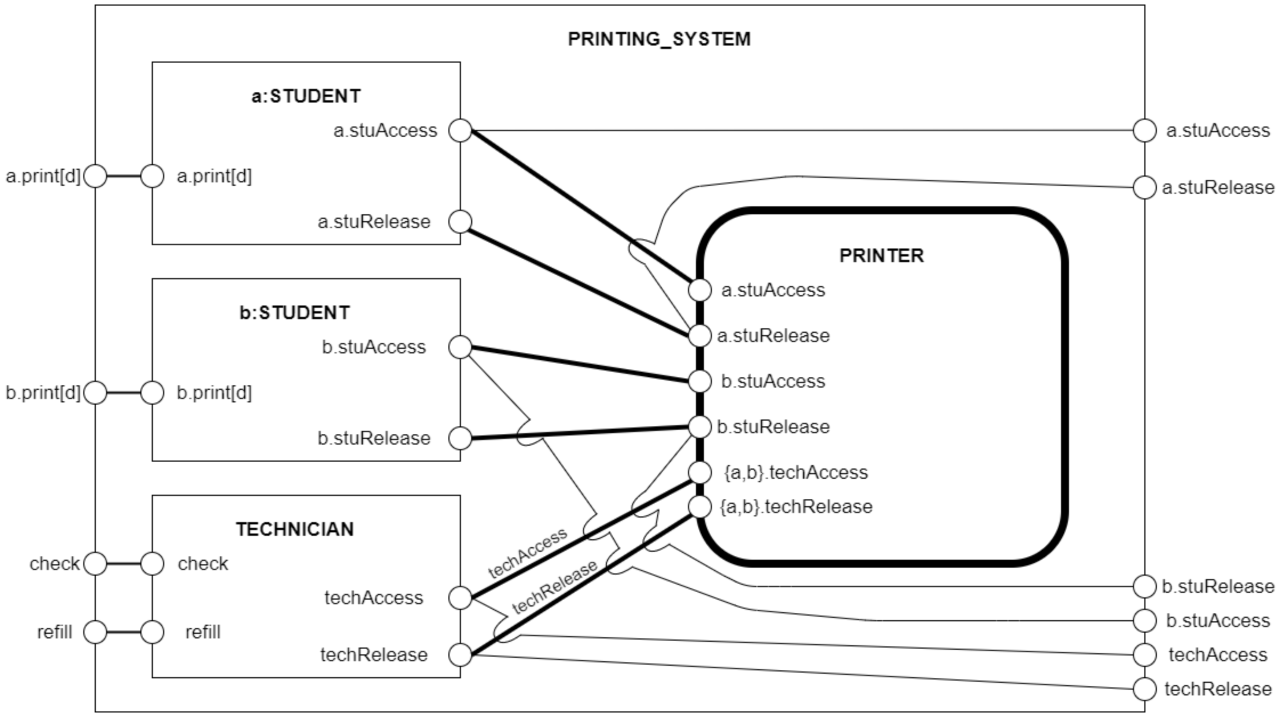
(Add rows as necessary.)

Synchronous Actions	Synchronised by Sub-Processes (List)
a.stuAccess	a:STUDENT, PRINTER
a.stuRelease	a:STUDENT, PRINTER
b.stuAccess	b:STUDENT, PRINTER
b.stuRelease	b:STUDENT, PRINTER
techAccess	TECHNICIAN, PRINTER
techRelease	TECHNICIAN, PRINTER

Sub-Process	Asynchronous Actions (List)
a:STUDENT	a.print[1..3]
b:STUDENT	b.print[1..2]
TECHNICIAN	refill, check

5. Parallel Composition Structure Diagram

The structure diagram for the parallel composition.


PRINTING_SYSTEM
a:STUDENT
a.stuAccess
a.print[d]
a.print[d]
a.stuRelease
a.stuAccess
a.stuRelease
PRINTER
a.stuAccess
a.stuRelease
b.stuAccess
b.stuRelease
{a,b}.techAccess
{a,b}.techRelease
b:STUDENT
b.stuAccess
b.print[d]
b.print[d]
b.stuRelease
TECHNICIAN
check
check
techAccess
refill
refill
techRelease
techAccess
techRelease
b.stuRelease
b.stuAccess
techAccess
techRelease