6SENG002W Concurrent Programming

FSP Process Composition Analysis & Design Form

Name	Shiromi Thevarajan
Student ID	2018117 / w1714893
Date	01/01/2022

1. FSP Composition Process Attributes

Attribute	**Value**
Name	BANKING_SYSTEM
Description	The BANKING_SYSTEM is a composite process, which includes five primitive processes CURRENT_ACCOUNT, STUDENT, GRANDMOTHER, LOAN_COMPANY and UNIVERSITY. This composite process models a banking system. And the users (student, grandmother, loan company and university) have mutually exclusive access to the shared current account when operating on it.
Alphabet	{ grandmother.currentAccount.computeNewBalance[3..7], grandmother.currentAccount.readBalance[5], grandmother.currentAccount.updateBalance[3..7], grandmother.depositBirthdayMoney[1..2], grandmother.sendEBirthdayCard, loanCompany.currentAccount.computeNewBalance[3..7], loanCompany.currentAccount.readBalance[5], loanCompany.currentAccount.updateBalance[3..7], loanCompany.depositLoanAmount[1..2], student.buySamsungPhone, student.currentAccount.computeNewBalance[3..7], student.currentAccount.readBalance[5], student.currentAccount.updateBalance[3..7],

	student.withdrawMoney[1..2], university.currentAccount.computeNewBalance[3..7], university.currentAccount.readBalance[5], university.currentAccount.updateBalance[3..7], university.withdrawFees[1..2] }
Sub-processes	STUDENT, GRANDMOTHER, LOAN_COMPANY, UNIVERSITY, CURRENT_ACCOUNT
Number of States	41
Deadlocks	No
Deadlock Trace(s)	N/A

2. FSP "main" Program Code

FSP Program:

```
/* CONSTANTS */
const MIN_TRANSACTION_AMOUNT  = 1
const MAX_TRANSACTION_AMOUNT = 2

/* RANGES */
range TRANSACTION_AMOUNT =
MIN_TRANSACTION_AMOUNT..MAX_TRANSACTION_AMOUNT
range INITIAL_BALANCE           = 5..5
range NEW_BALANCE               = 3..7

/* SETS */
set AccountUsers    = { student, grandmother, loanCompany, university }
set AccountActions = { currentAccount.{ readBalance [ INITIAL_BALANCE ] ,
computeNewBalance [ NEW_BALANCE ] , updateBalance [ NEW_BALANCE ] } }

/* BANKING SYSTEM PROCESS */
|| BANKING_SYSTEM = (
        student : STUDENT ||
        grandmother : GRANDMOTHER ||
        loanCompany : LOAN_COMPANY ||
        university : UNIVERSITY ||
        AccountUsers :: currentAccount : CURRENT_ACCOUNT ) .} .
```

3. Combined Sub-processes

Process	Description
STUDENT	The primitive process STUDENT represents the owner of the current account. This process includes actions withdrawMoney and buySamsungPhone. The withdrawMoney action withdraws money from the student’s bank account. And the buySamsungPhone action buys a Samsung phone. The STUDENT process synchronizes with CURRENT_ACCOUNT on readBalance, computeNewBalance and updateBalance actions.
GRANDMOTHER	The primitive process GRANDMOTHER represents the grandmother of the student. This process includes actions depositBirthdayMoney and sendEBirthdayCard. The depositBirthdayMoney action deposits birthday present money into the student's bank account. And the sendEBirthdayCard action sends an e-birthday to the student. The GRANDMOTHER process synchronizes with CURRENT_ACCOUNT on readBalance, computeNewBalance and updateBalance actions.
LOAN_COMPANY	The primitive process LOAN_COMPANY represents the loan company that pays for the education of the student. This process includes the action depositLoanAmount. The depositLoanAmount action deposits the loan amount into the student's bank account. The LOAN_COMPANY process synchronizes with CURRENT_ACCOUNT on readBalance, computeNewBalance and updateBalance actions.
UNIVERSITY	The primitive process UNIVERSITY represents the university that the student attends. This process includes action withdrawFees. The withdrawFees action withdraws course fees from the student's bank account. The UNIVERSITY process synchronizes with CURRENT_ACCOUNT on readBalance, computeNewBalance and updateBalance actions.
CURRENT_ACCOUNT	The primitive process CURRENT_ACCOUNT represents the bank account of the student. And this is a shared resource of the banking system. This process includes readBalance, computeNewBalance and

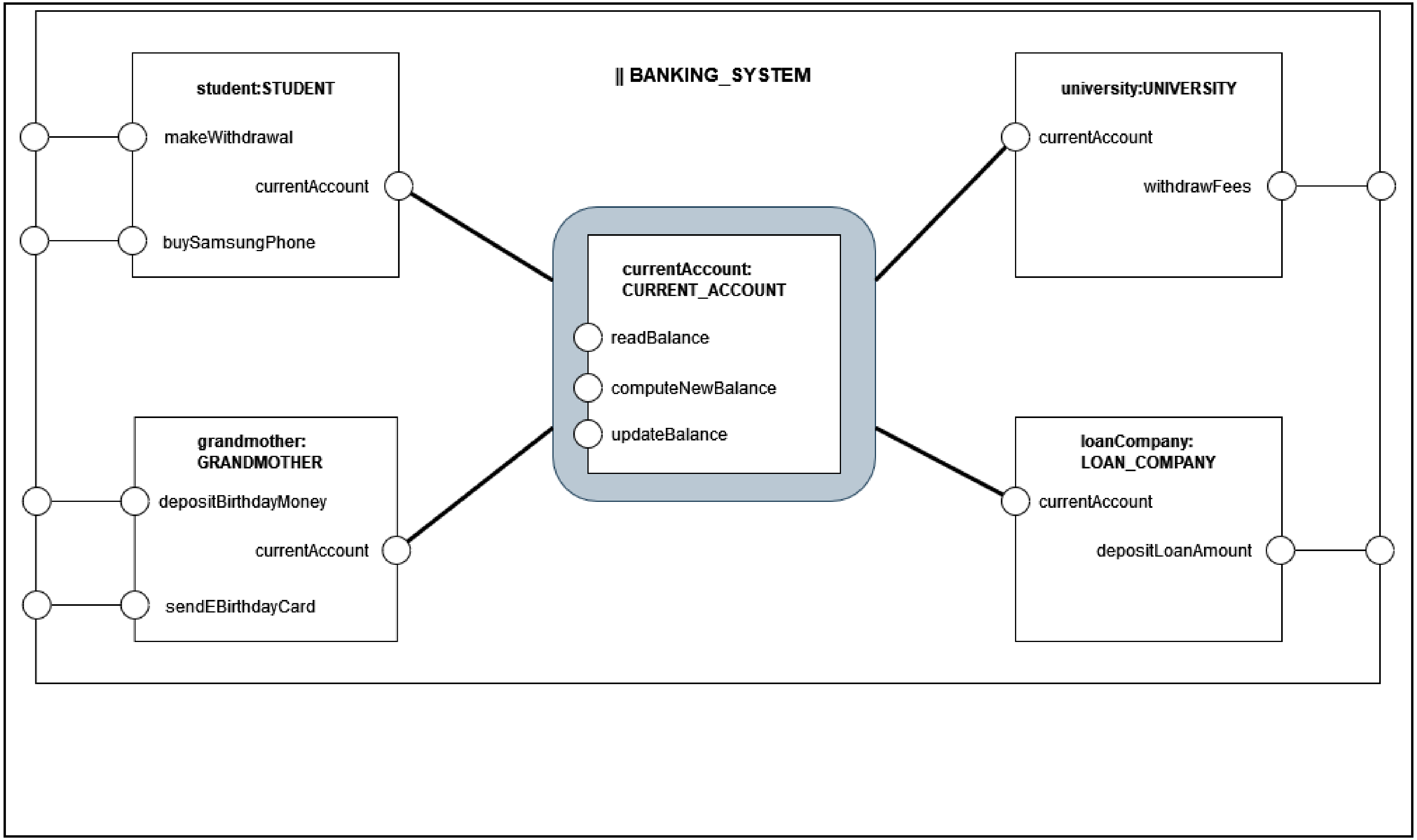
	updateBalance actions. The readBalance action reads the current balance of the bank account. The computeNewBalance action calculates the new balance after a transaction is performed. And the updateBalance action updates the current balance of the account with the new balance.

4. Analysis of Combined Process Actions

Synchronous Actions	Synchronised by Sub-Processes
student.currentAccount.readBalance[5], student.currentAccount.computeNewBalance[3..7], student.currentAccount.updateBalance[3..7]	STUDENT, CURRENT_ACCOUNT
grandmother.currentAccount.readBalance[5], grandmother.currentAccount.computeNewBalance[3..7], grandmother.currentAccount.updateBalance[3..7]	GRANDMOTHER, CURRENT_ACCOUNT
loanCompany.currentAccount.readBalance[5], loanCompany.currentAccount.computeNewBalance[3..7], loanCompany.currentAccount.updateBalance[3..7]	LOAN_COMPANY, CURRENT_ACCOUNT
university.currentAccount.readBalance[5], university.currentAccount.computeNewBalance[3..7], university.currentAccount.updateBalance[3..7]	UNIVERSITY, CURRENT_ACCOUNT

Sub-Process	Asynchronous Actions
STUDENT	student.withdrawMoney, student.buySamsungPhone
GRANDMOTHER	grandmother.depositBirthdayMoney, grandmother.sendEBirthdayCard
LOAN_COMPANY	loanCompany.depositLoanAmount
UNIVERSITY	university.withdrawFees

5. Parallel Composition Structure Diagram


|| BANKING_SYSTEM
student:STUDENT
makeWithdrawal
currentAccount
buySamsungPhone
university:UNIVERSITY
currentAccount
withdrawFees
currentAccount:
CURRENT_ACCOUNT
readBalance
computeNewBalance
updateBalance
grandmother:
GRANDMOTHER
depositBirthdayMoney
currentAccount
sendEBirthdayCard
loanCompany:
LOAN_COMPANY
currentAccount
depositLoanAmount