Modeling the Educational System using Service-Oriented Modeling and Aspect-Oriented Technology

By Nguyen Thi Hong Nhung

A thesis submitted to
School of Information Science,
Japan Advanced Institute of Science and Technology
in partial fulfillment of the requirements
for the degree of
Master of Information Science
Graduate Program in Information Science

Written under the direction of
Professor Koichiro Ochimizu

March, 2011

Acknowledgements

First and foremost, I would like to express my deepest thanks to my supervisor – Professor Koichiro Ochimizu for his guidance through the study program in JAIST, Japan. His valuable advices and support have helped me to overcome many difficulties in the studying and researching during my study in JAIST. He has made me interested in science and doing research. My study could not have been completed without him.

Secondly, I would like to acknowledge TIS Inc. for the financial support on the study of Master program in JAIST, Japan.

Thirdly, I would like to acknowledge my friends, colleagues at JAIST who have been beside me and helped me a lot in life in JAIST. Besides, I also would like to express my gratefulness to all staffs at JAIST who have helped me a lot during the study time in JAIST, Japan.

Finally and most importantly, I would like to thank my family for their unconditional love and encouragement through my study program and throughout my life.

Nguyen Thi Hong Nhung,

JAIST, Japan, March 2011.

Abstract

All schools provide many services to the students. Some schools have simple educational systems, but some others have quite complex systems where there are so many services and they are heavily related to each other. Such type of educational systems do not support for students to be able to easily find and choose services which are suitable for them. Besides, another problem is the modifiability of the educational system. Almost every year the school's curriculum is revised, so it is better to have a modifiable structure for the system. Therefore, our aim is to organize the services as a single model which is more understandable and useful for the students and modifiable for the schools.

In this research, we take the **Japan Advanced Institute of Science and Technology (JAIST) educational system (JES)** as an example, analyze the system, model the services through Service-Oriented Modeling and deal with relationship between services as well as crosscutting concerns by Aspect-Oriented Technology.

Table of Contents

Acknowledgements 1

Abstract 2

Chapter 1 - Introduction 7

1.1 Problem in Educational Systems 7

1.2 Research Purpose 8

Chapter 2 - Background 9

2.1 Service-Oriented Technology 9

2.1.1 Service-Oriented Architecture 9

2.1.2 Service-Oriented Modeling 10

2.1.3 Business Process Modeling 10

2.2 Aspect-Oriented Technology 11

Chapter 3 – Basic Model of Education System 12

3.1 Basic Model of ES 12

Chapter 4 - Service-Oriented Modeling 18

4.1 Model of Educational System 18

4.2 Type of Service selections by Students 19

4.3 Interaction between Student and School actors 21

4.4 Business Process Modeling 27

Chapter 5 - Aspects 32

5.1 Type of Aspects 32

5.2 Constraint Aspects 32

5.2.1 Definition of Constraints 32

5.2.2 Example of Constraints 33

5.2.3 Definition of Constraint Aspects 34

5.2.4 Structure of Constraint Aspects 35

5.3 Relationship Aspects 40

5.3.1 Problem in ESs 40

5.3.2 Definition of Relationship Aspects 46

5.3.3 Structure of Relationship Aspects. 47

5.4 Extension Aspects 49

5.4.1 Definition of Extension Aspects 49
5.4.2 Example of Extension Aspects 49
5.4.3 Structure of Extension Aspects. 50
5.5 General model of educational system with Aspects 51
5.6 General Concept Classification 52
Chapter 6 - Conclusion and Future work 55
Bibliography 56

List of Figures

Figure 1: JAIST provides many services to the students. 7
Figure 2: Research's purpose. 8
Figure 3: Elements of a SOA [2]. 9
Figure 4: The service-oriented modeling and architecture method [3]. 10
Figure 5: Example of BPM of a process with a normal flow with the BPMN [4]. 10
Figure 6: Figure Editor in UML [5]. 11
Figure 7: Aspect-Oriented Figure Editor in UML [5]. 11
Figure 8: Structure of the JES. 12
Figure 9: Some educational services provided by JAIST. 13
Figure 10: Degree Program service provided by JAIST. 13
Figure 11: Example of Top-level educational service. 14
Figure 12: Course service provided by JAIST. 15
Figure 13: Subject service provided by JAIST. 16
Figure 14: Teacher's knowledge units that are used in Subjects. 17
Figure 15: Model of Educational System. 18
Figure 16: General selections from Students. 21
Figure 17: Scenario of M2 program in JES. 22
Figure 18: Program service part is the general model of ES. 27
Figure 19: Example of BPM for Program service. 27
Figure 20: Course service part in the general model of ES. 28
Figure 21: Example of BPM for Course service. 28
Figure 22: Research service part in the general model of ES. 29
Figure 23: Example of BPM for Research service. 29
Figure 24: Subject service part in the general model of ES. 30
Figure 25: Example of BPM for [451] Subject service. 30
Figure 26: Example of BPM for [226] Subject service. 31
Figure 27: Constraint part in the general model of ES. 32
Figure 28: Example of Constraints in JES. 33
Figure 29: Example of Constraints in BPM. 34
Figure 30: Example of Constraints in BPM. 34
Figure 31: Structure of CA. 35
Figure 32: Structure of "CheckRFC" Aspect. 35
Figure 33: Example of Constraint Aspects in BPM. 36
Figure 34: Structure of "CheckMRFC" Aspect. 39
Figure 35: Structure of "CheckDRFC" Aspect. 39
Figure 36: Relationship part in the general model of ES. 40
Figure 37: Relationship between Subjects in JES. 45
Figure 38: S1 and S3 have relationships with S2. 46
Figure 39: Information about relationships is stored in each subject. 46

Figure 40: Structure of RA. 47
Figure 41: Example of Relationship Aspects & their structure. 47
Figure 42: JAIST and Teacher Services cut crossing. 49
Figure 43: Example of Teacher's Services crosscut JAIST's Services. 50
Figure 44: Structure of EA. 50
Figure 45: Structure of example EA in JES. 50
Figure 46: Example of EA in UML. 51
Figure 47: The general model of educational system (with Aspects). 52

List of Tables

Table 1: Table of Programs and RFCs in JES. 36
Table 2: Table of Subjects and Prerequisites in JES. 40
Table 3: Example of Relationship Aspects in JES. 48
Table 4: Table of the general concept classification. 53

Chapter 1 - Introduction

1.1 Problem in Educational Systems

All schools provide many services to the students. Some schools have simple educational systems (ESs), but some others have quite complex systems where there are so many services and they are heavily related to each other. Such type of educational systems do not support for students to be able to easily find and choose services which are suitable for them. Besides, another problem is the modifiability of the educational system. Almost every year the school's curriculum is revised, so it is better to have a modifiable structure for the system. Therefore, our aim is to organize the services as a single model which is more understandable and useful for the students and modifiable for the schools.

Figure 1: JAIST provides many services to the students.

For example, JAIST is an advanced institute which provides many services to the students, such as: Degree Programs, Courses, Subjects, Seminars, etc. It is highly recommended to define a model of the JAIST educational system and enable the students to understand and choose suitable services.

1.2 Research Purpose

My research is aimed at modeling the School's educational system using Service-Oriented Modeling and Aspect-Oriented Technology.

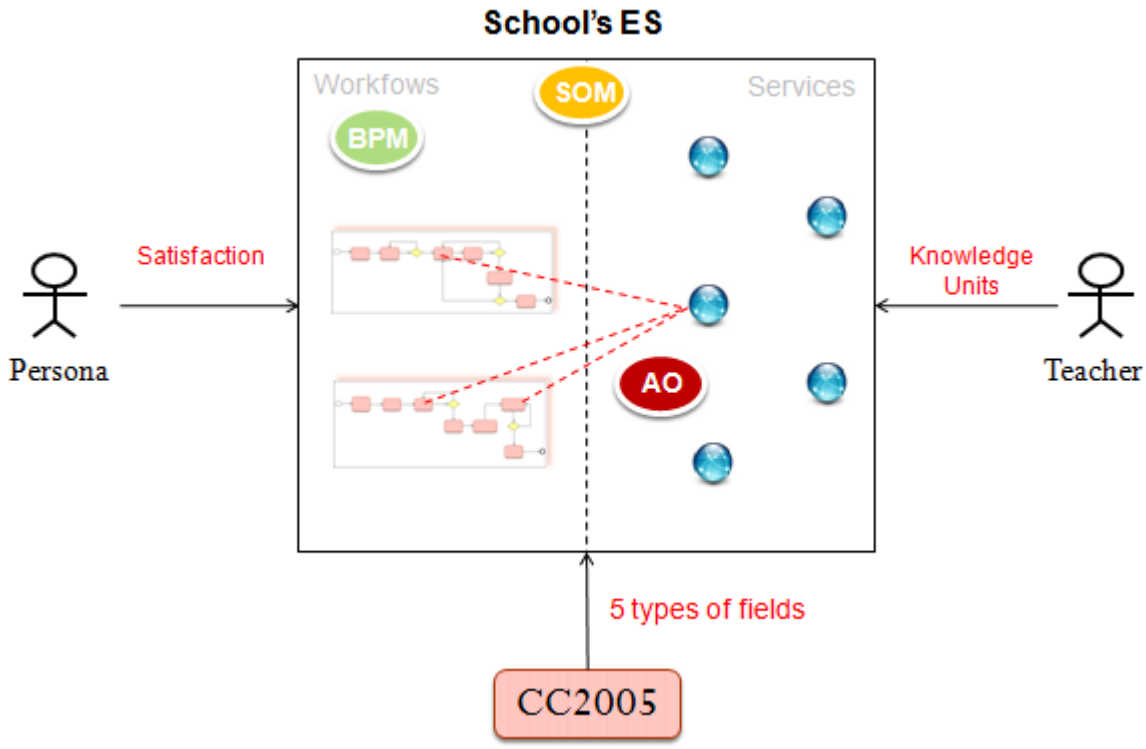


Figure 2: Research's purpose.

The primary goal of my research includes:

- Using SOM to model the system services.
- Describe several services through BPM.
- Build the general model for the educational system.
- Analyze the system to find crosscutting concerns and define Aspects which are specific to them using AO Technology.
- Define a new model which includes Aspects.

Chapter 2 - Background

2.1 Service-Oriented Technology

2.1.1 Service-Oriented Architecture

A **Service-Oriented Architecture (SOA)** is essentially a collection of services which communicate with each other. The communication can involve either simple data passing or it could involve two or more services coordinating some activity [1].

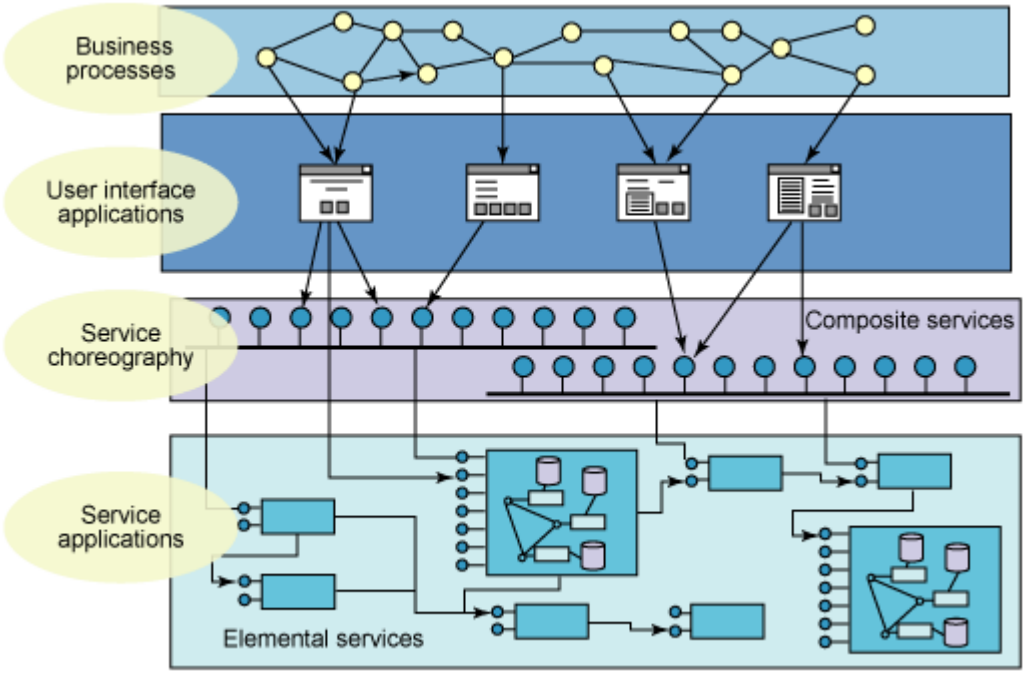


Figure 3: Elements of a SOA [2].

The figure above shows the elements of a SOA which includes: Business processes, User Interface applications, Service choreography and Service applications.

2.1.2 Service-Oriented Modeling

Service-Oriented Modeling (SOM) is the designing and specifying of service-oriented business systems within a SOA [3], or in other words, is the analysis and design of services. SOM typically aims to create models that provide a comprehensive view of the analysis, design, and architecture of all Entities in an organization.

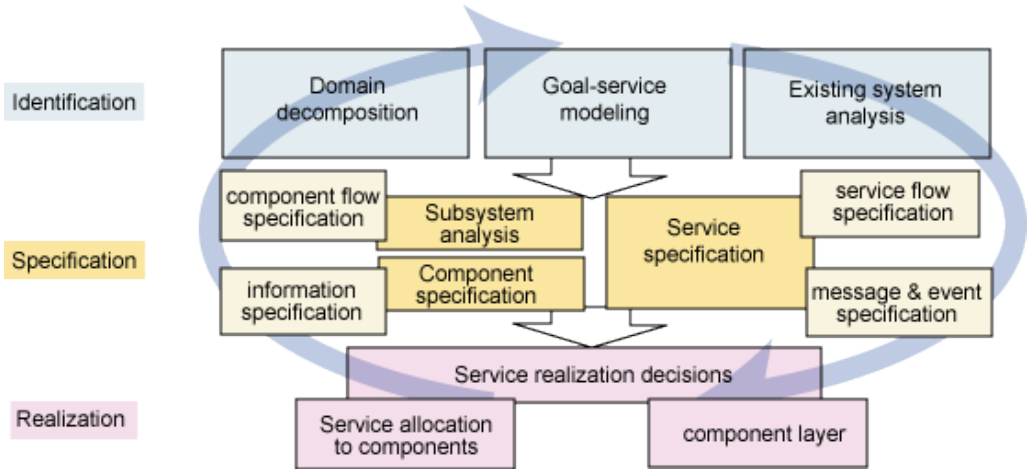


Figure 4: The service-oriented modeling and architecture method [3].

The figure above shows the process of service-oriented modeling and architecture which consists of three general steps: identification, specification and realization of services, components and flows of services.

2.1.3 Business Process Modeling

Business Process Modeling (BPM) is a key technology which represents the process activities of a system. **Business Process Modeling Notation (BPMN)** is a de-facto standard technology for BPM. It provides a graphical notation for specifying business processes in a **Business Process Diagram (BPD)**, based on flowcharting technique [4].

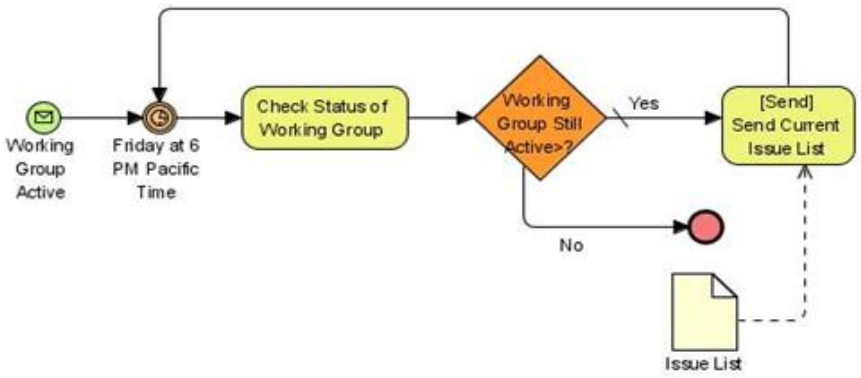


Figure 5: Example of BPM of a process with a normal flow with the BPMN [4].

2.2 Aspect-Oriented Technology

In the system, besides the main functionalities (or services), there are also other "crosscutting concerns" which crosscut across several services. In order to deal with them, to identify, and then to separate those concerns from the main services, we need to use a special technology. **Aspect-oriented Software Development (AOSD)** is an emerging technology that provides the advanced separating and weaving of crosscutting concerns by modularizing them into modularized concerns which are called **Aspects** [5].

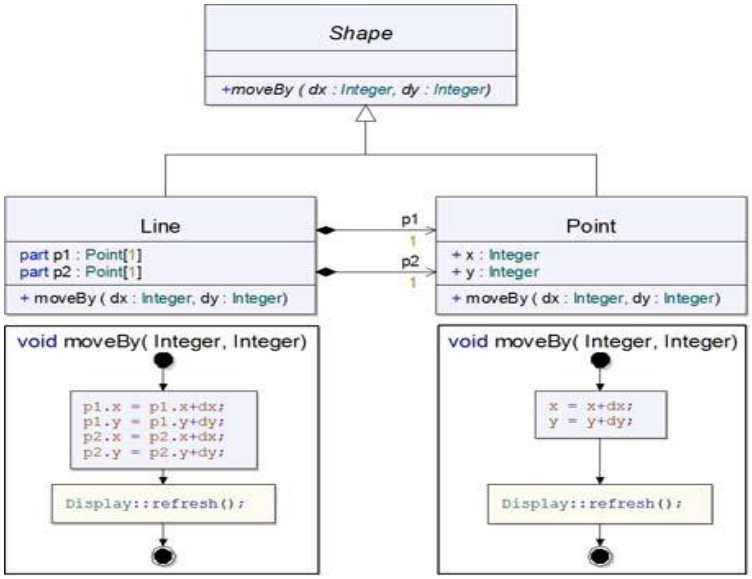


Figure 6: Figure Editor in UML [5].

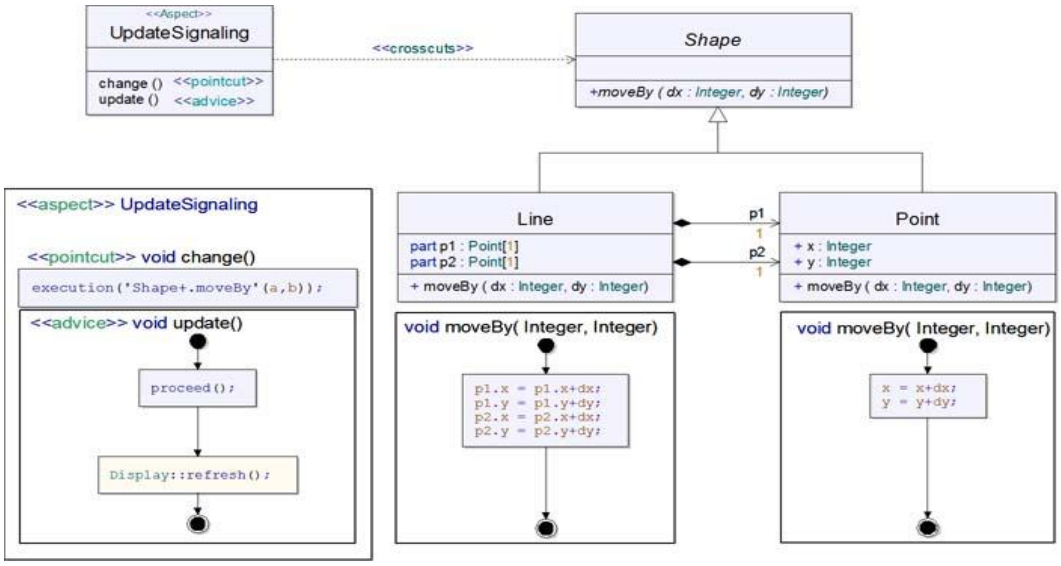


Figure 7: Aspect-Oriented Figure Editor in UML [5].

By analyzing the system, we will find and define Aspects that are applied to this research.

Chapter 3 – Basic Model of Education System

3.1 Basic Model of ES

Each school has a model of the educational system which includes many services, actors, classes, etc. In Figure 8, we show an example of the structure of an educational system:

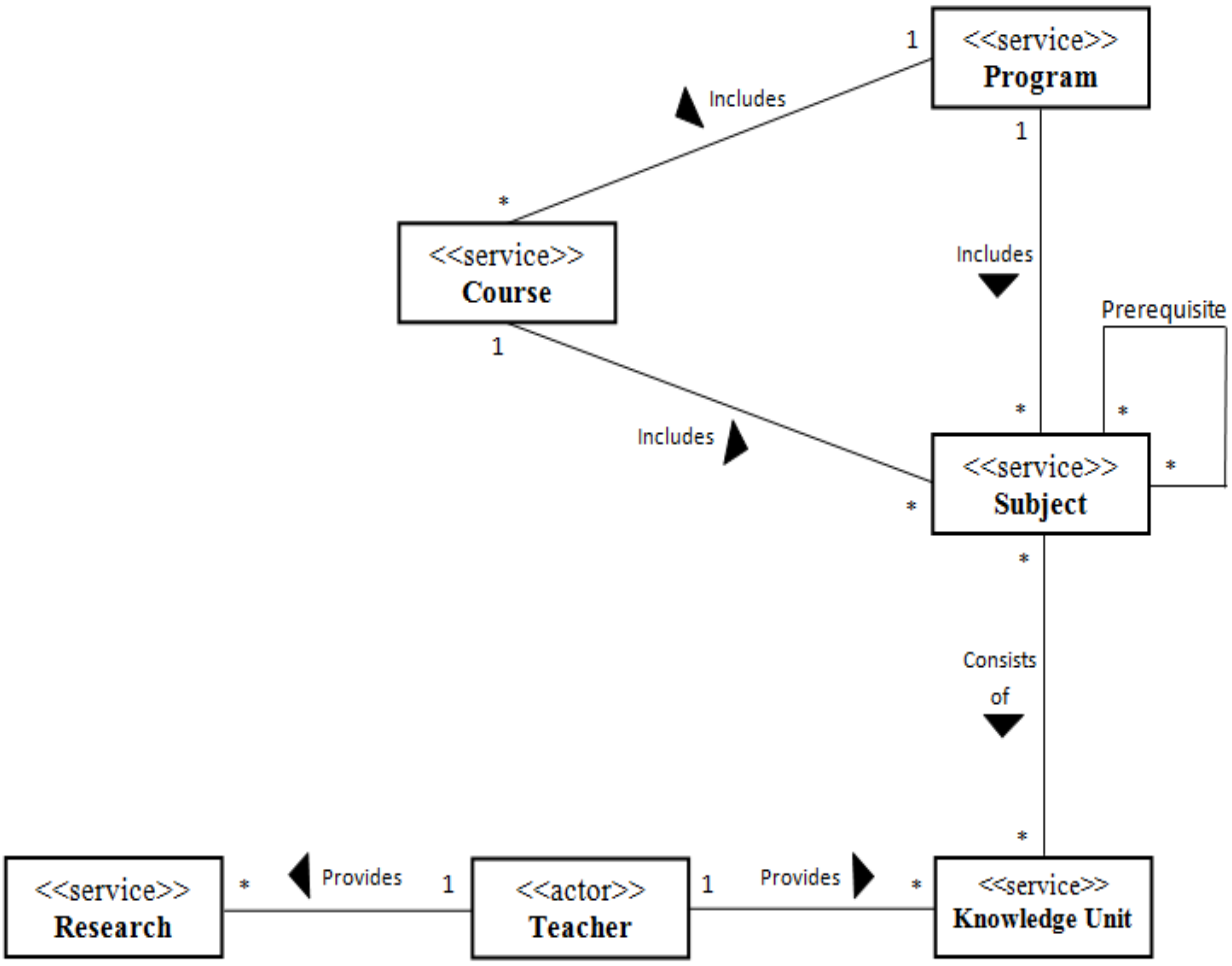


Figure 8: Structure of the JES.

According to Figure 8, we need to understand some concepts which are defined as follows:

- **Service:** is an organization‘s (School) or individual’s (Teacher) ability to provide its students with their wants and needs.

For example, JAIST provides many educational services, such as: Degree Program, Course, Subject, Seminar, Research, etc. as shown in the Figure 9:

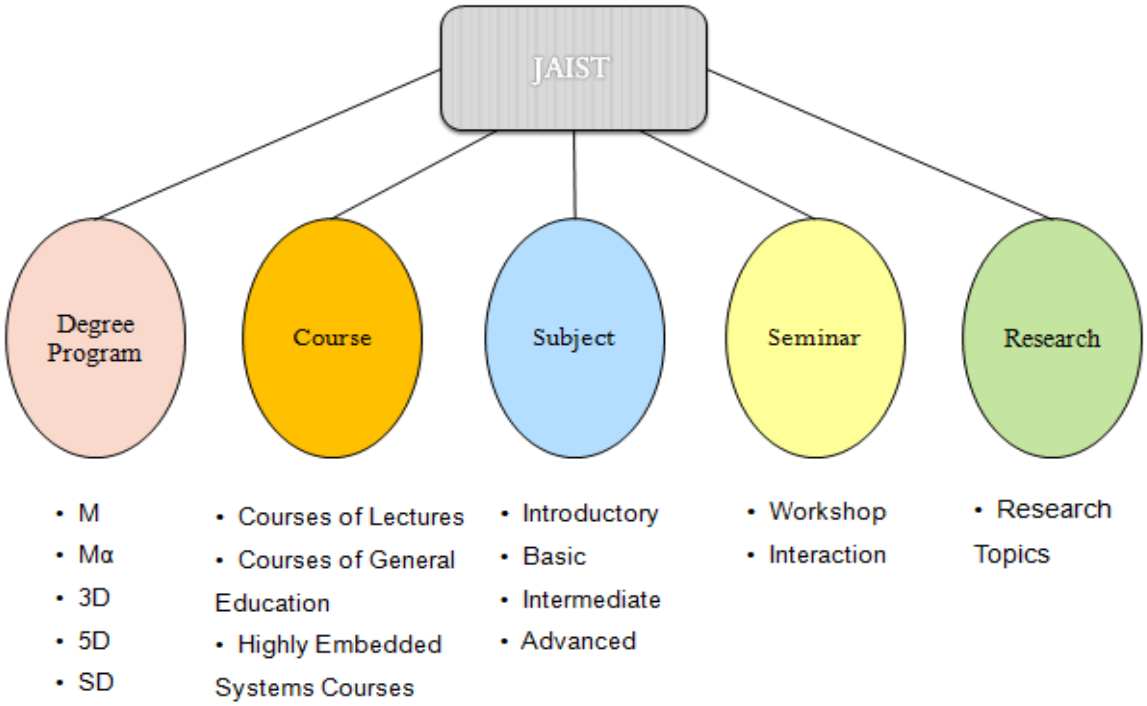


Figure 9: Some educational services provided by JAIST.

- **Degree Program service:** is a top-level educational service which includes all other educational services.

For example, JAIST Degree Program service provides five types of programs as follows:

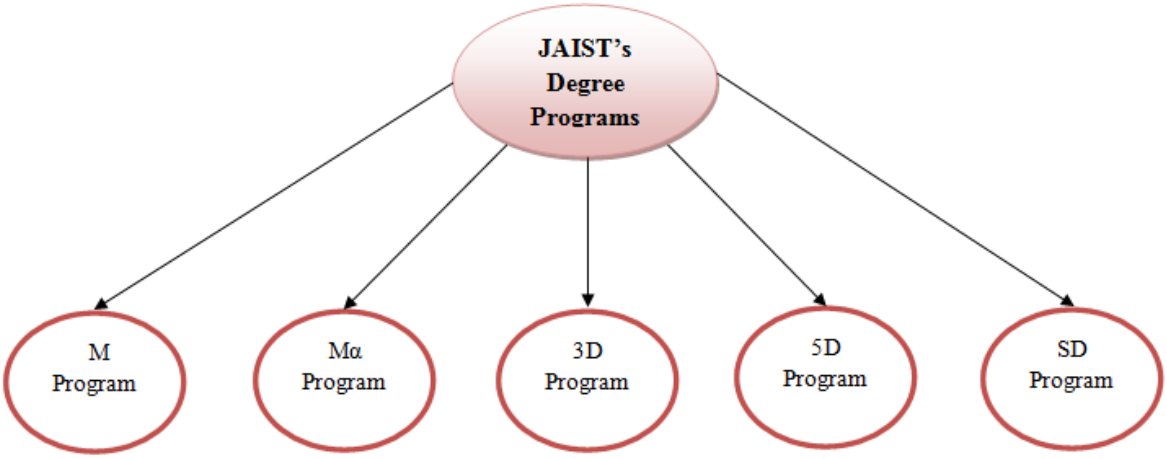


Figure 10: Degree Program service provided by JAIST.

And in Master Program, there can also be other services like Course with special Subjects, other Subjects, Seminars and Research as shown in Figure 11:

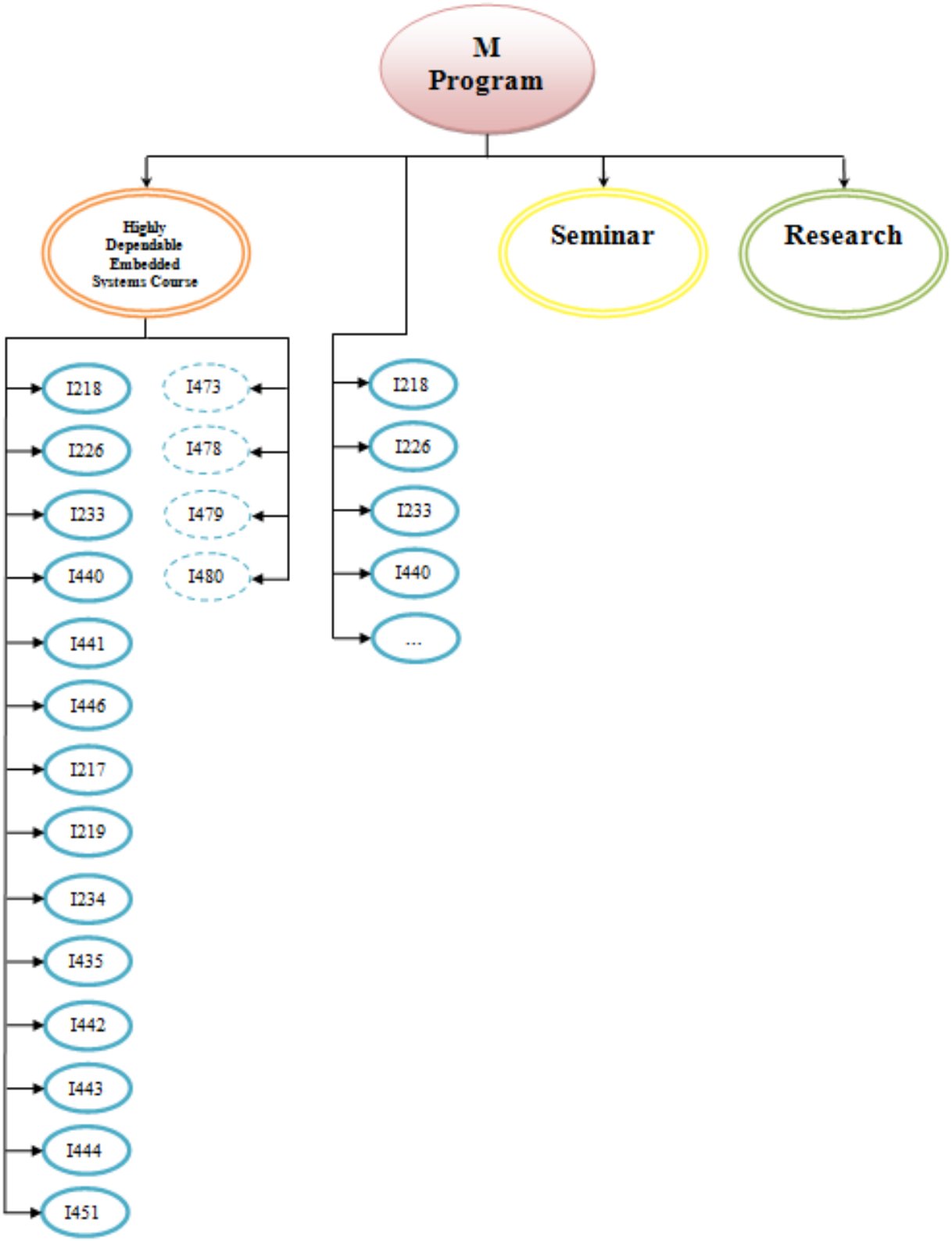


Figure 11: Example of Top-level educational service.

- **Course service:** is a special case of degree program which includes some special subjects.

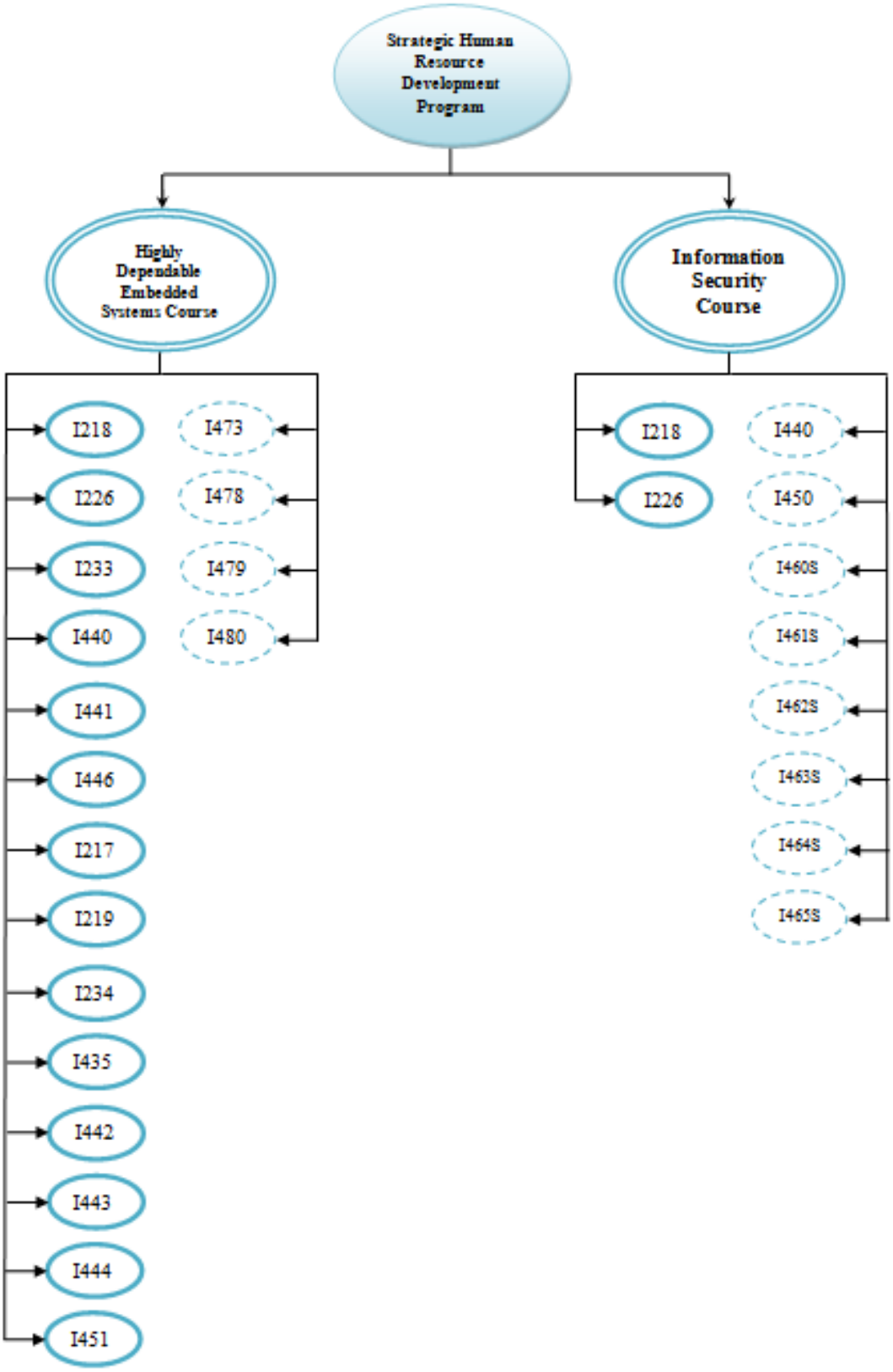


Figure 12: Course service provided by JAIST.

- **Subject service:** is a set of related knowledge units provided by Teachers to Students.

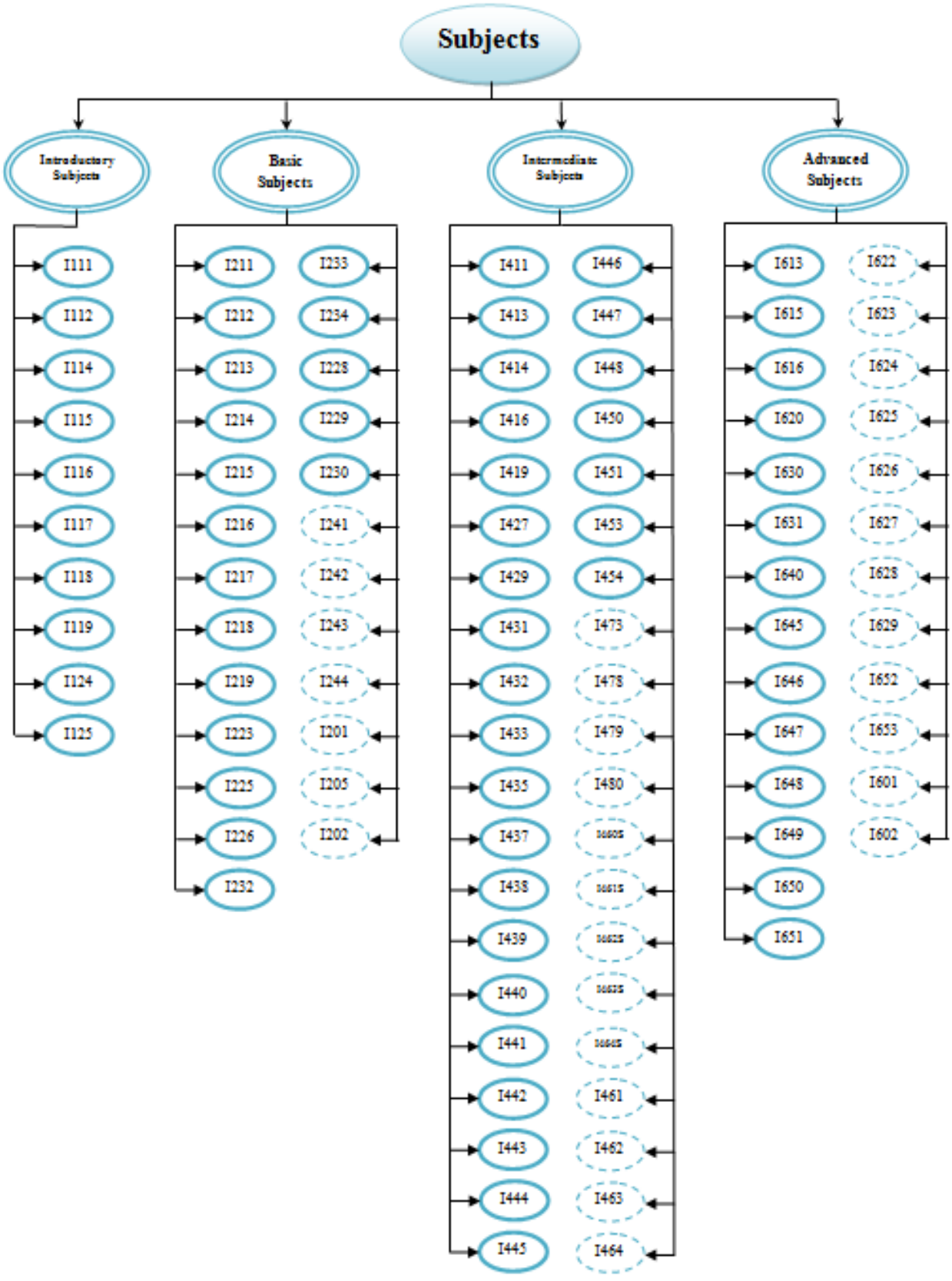


Figure 13: Subject service provided by JAIST.

- **Knowledge unit:** is a Teacher's knowledge.

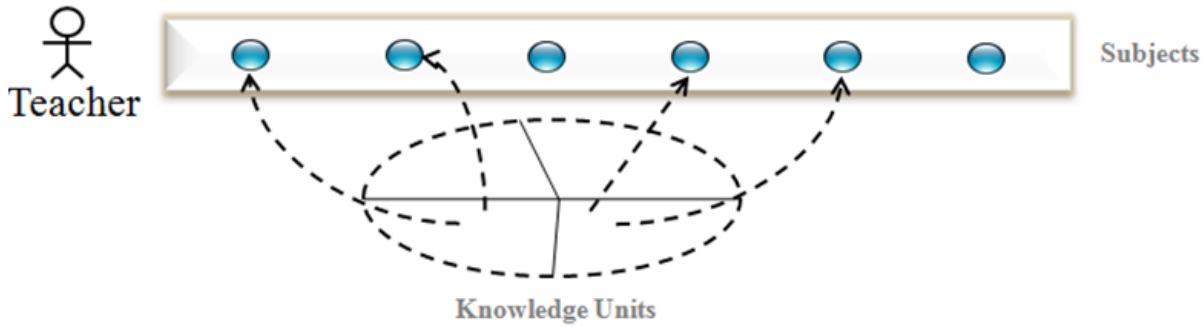


Figure 14: Teacher's knowledge units that are used in Subjects.

Each Teacher has a group of Knowledge Units, such as: knowledge about Object-Oriented Technology, knowledge about Aspect-Oriented Technology or about Databases, etc. And those knowledge units can be used in the same or different services (for example, Subjects) provided by Teacher.

Chapter 4 - Service-Oriented Modeling

4.1 Model of Educational System

The Figure 8 mentioned above represents an example of the basic model of educational system which includes the services provided by the School and Teacher actors. Students come to the school will use those services by performing many processes. And besides, there are many types of students who use those services. Therefore, in Figure 15, we add several more classes to the basic model:

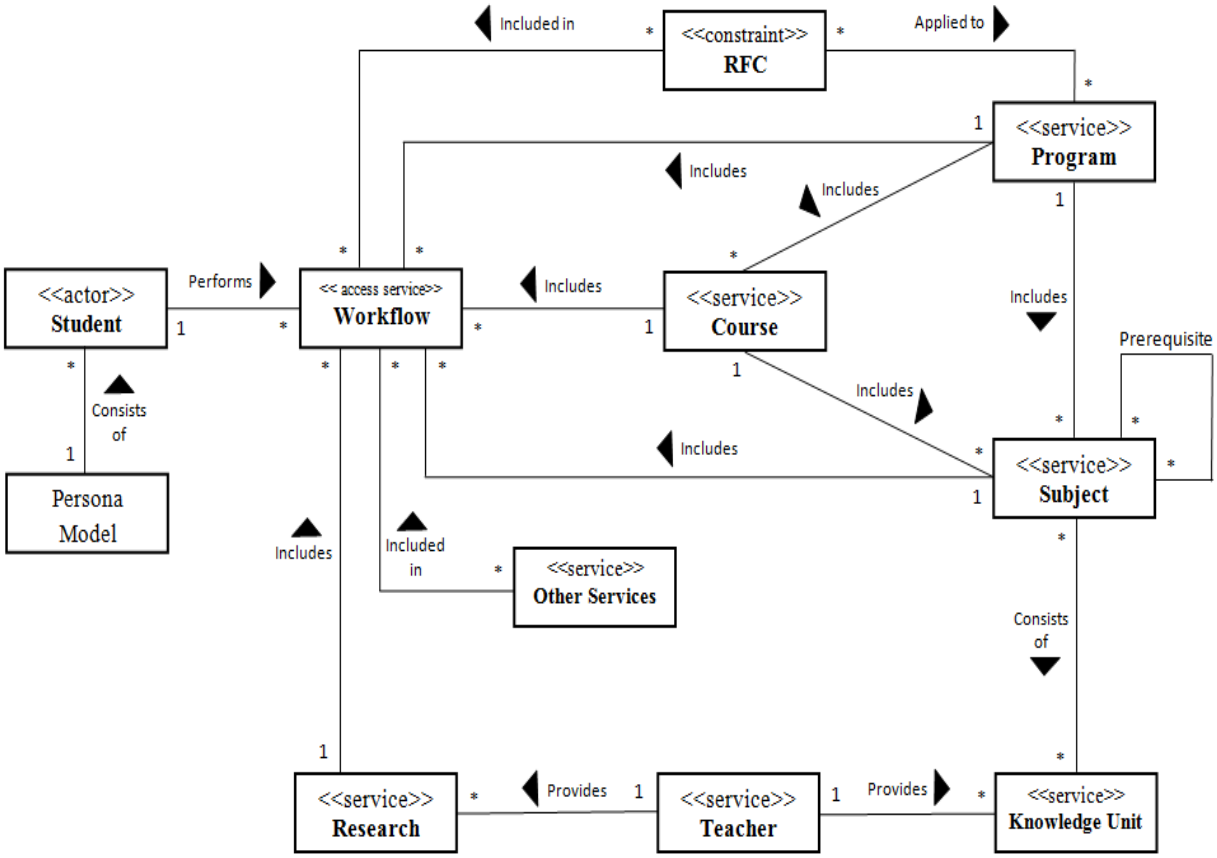


Figure 15: Basic structure of JES.

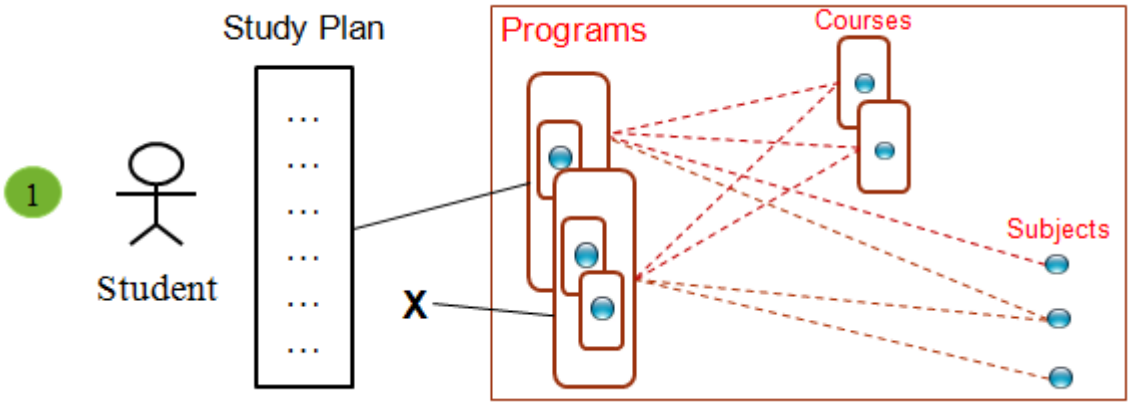
In Figure 15,

- **Workflow:** defines the way to access each service which will be described in Section 4.4 of this Chapter.
- **Persona Model:** represents the model of type of Student actors who use the services provided by School.
- **Constraint RFC (Requirements for Completion):** refers to the rules or requirements which are applied to each Program service and are also included in the Workflow.
- **Other services:** such as the guidance and support from the School, etc.

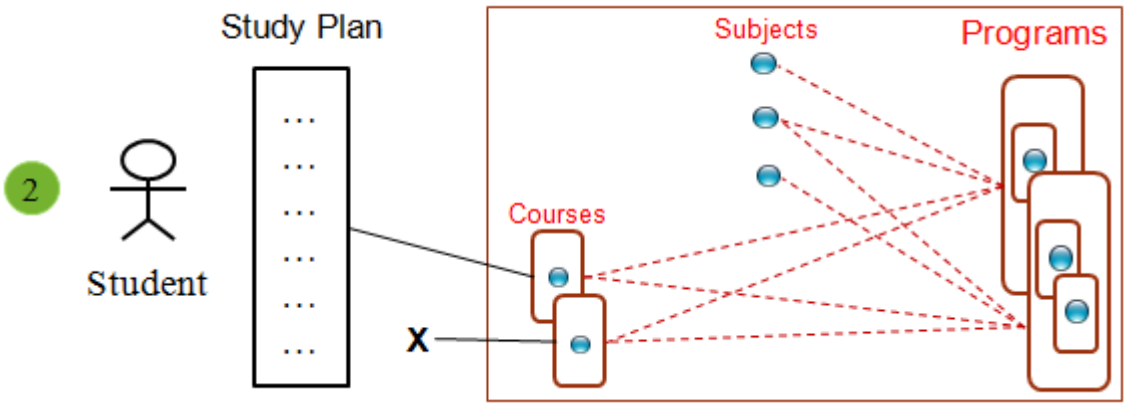
4.2 Type of Service selections by Students

The consideration of this section does not reflect to this thesis, but it is important for the building of a support system in the future.

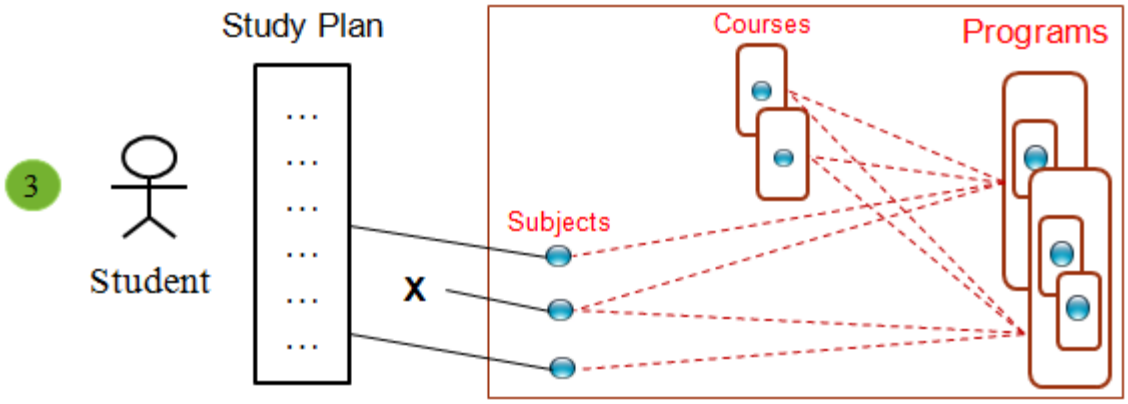
Students enter to the School, based on their study plans, can have some of service selection ways as follows:



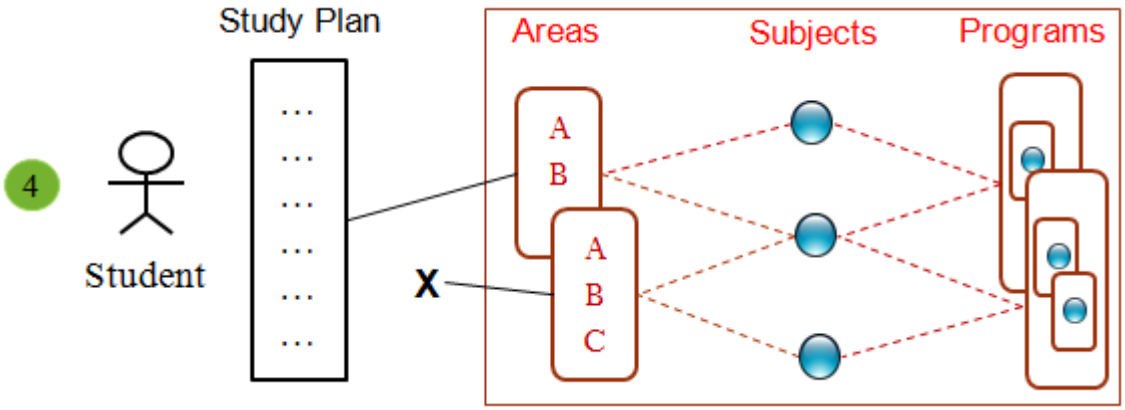
First, based on their study plan, students can choose their desired programs or courses provided by the School and then subjects provided by teachers.



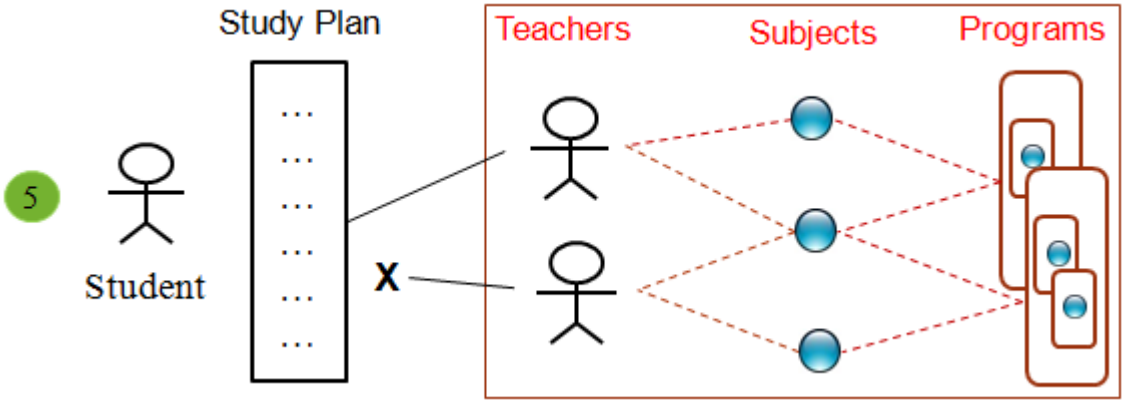
Or, students may choose their desired courses or subjects, and then programs.



Or, they may choose interested in subjects, then courses and programs.



Or, from the given areas, they choose suitable subjects, then programs.



Or, they may choose interested in teachers, with whom they want to do the researches with, then teachers' subjects, and then programs.

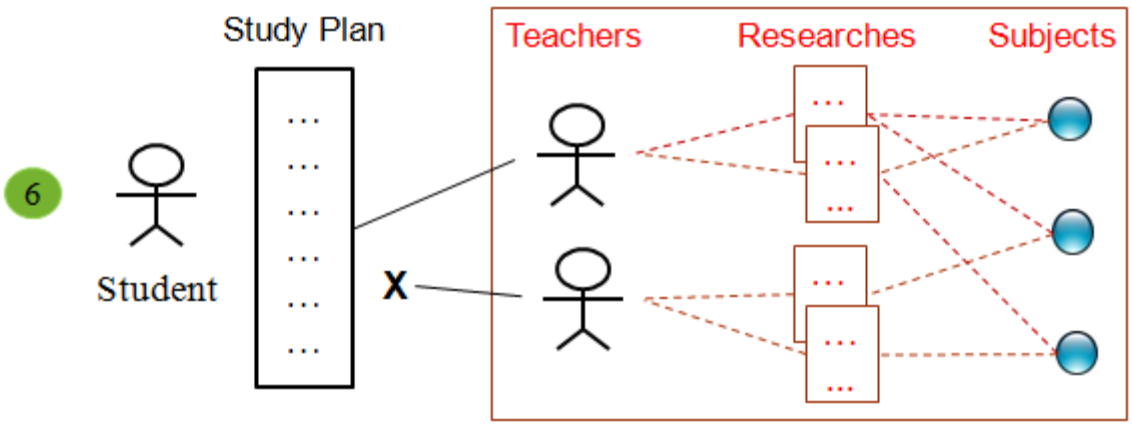


Figure 16: General selections from Students.

Or, teachers' researches, then their subjects.

The options illustrated above are just the general service selections that students may choose. By having these models, new students based on their study plans can know what services to choose and how to choose them before or when entering into the School.

4.3 Interaction between Student and School actors

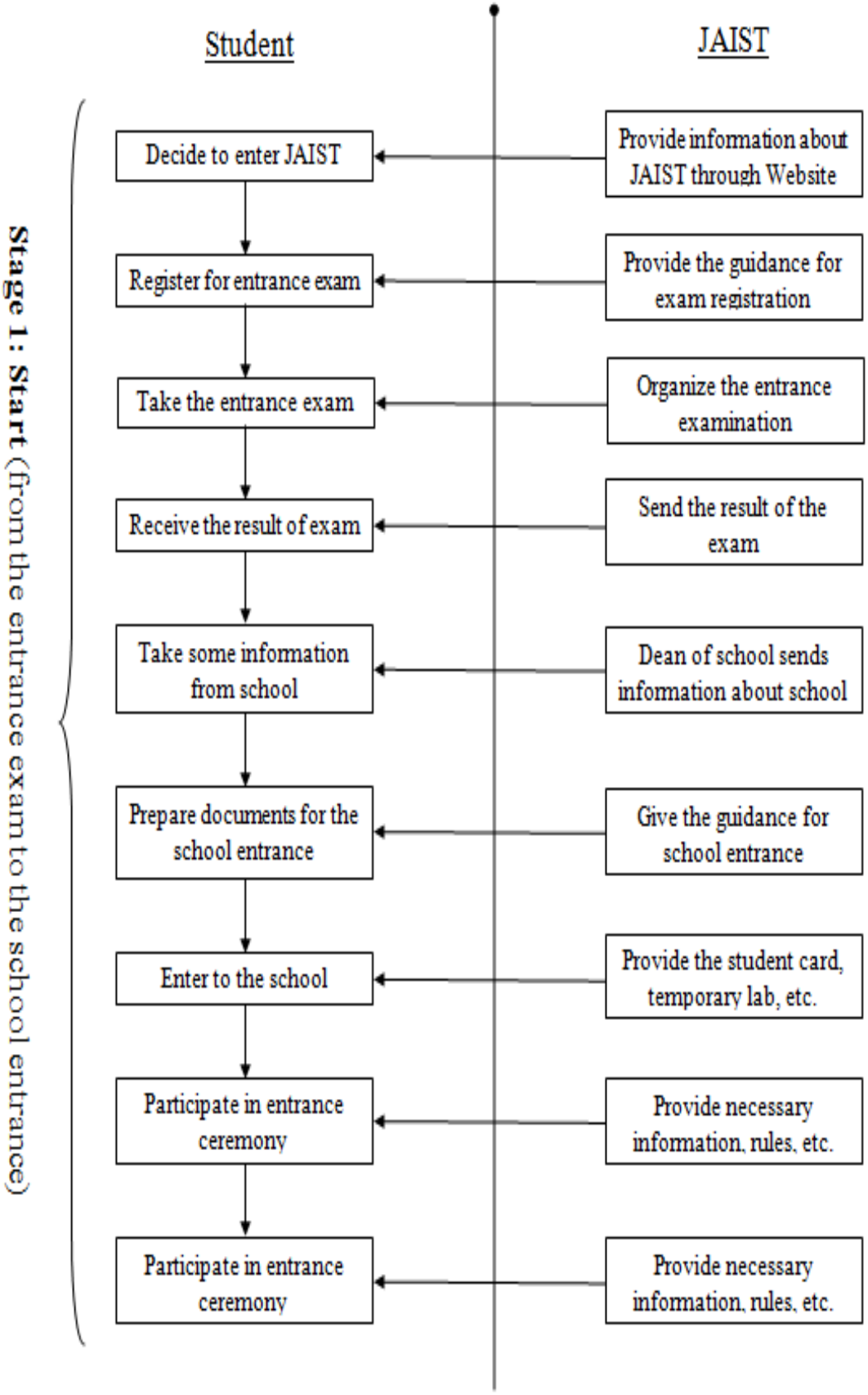
First, we are now going to talk about the interaction between "Workflow" and "Other Services".

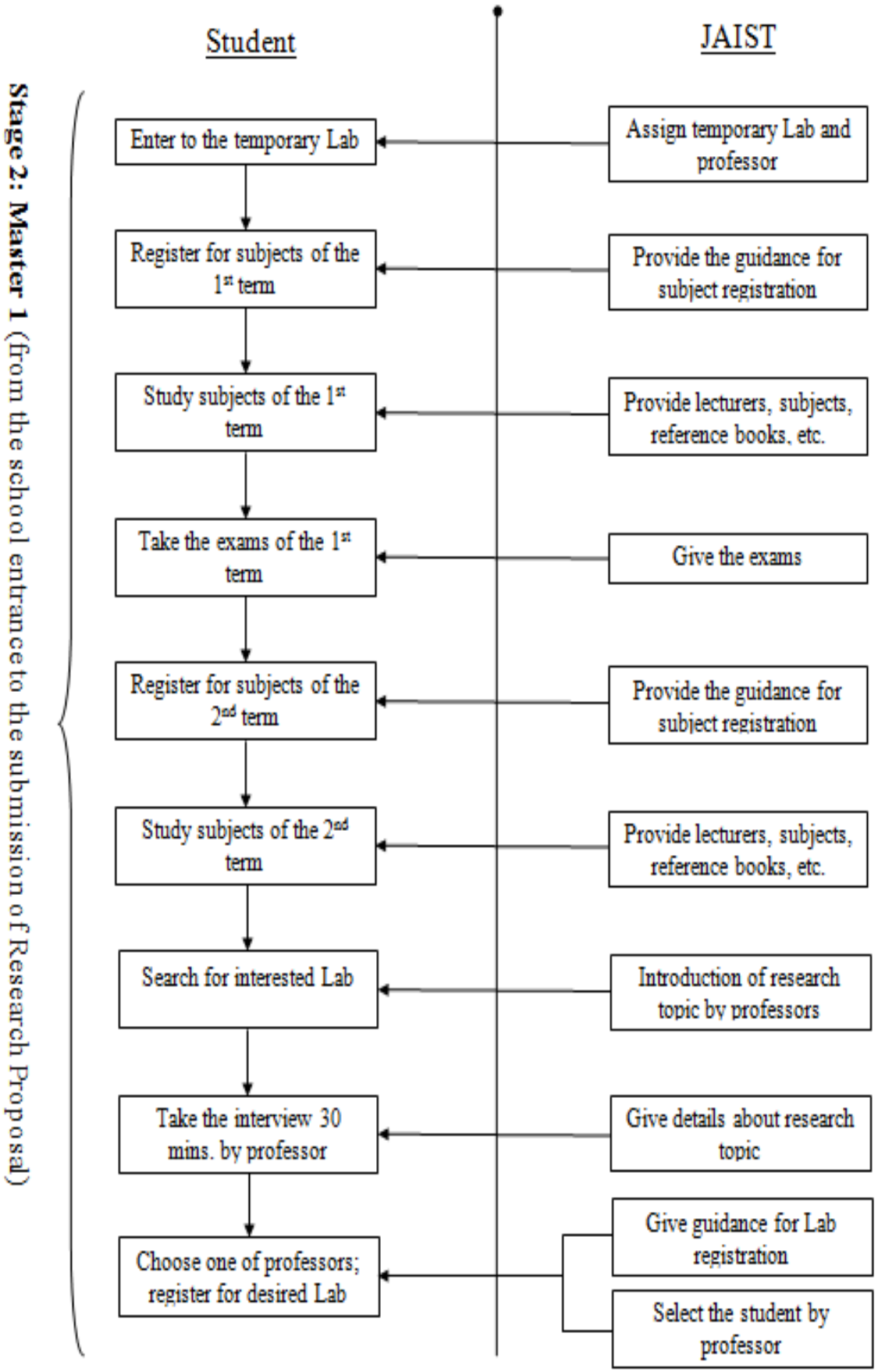
During the study years in the School, a student will perform many processes which interact with the services provided by the School. To understand about the interaction between Student and School actors, we use Scenarios. Figure 4 shows an example relationship between Scenario of accessing the M Program service and other services provided by JAIST which we made by using **"Theatre Service Model"**.

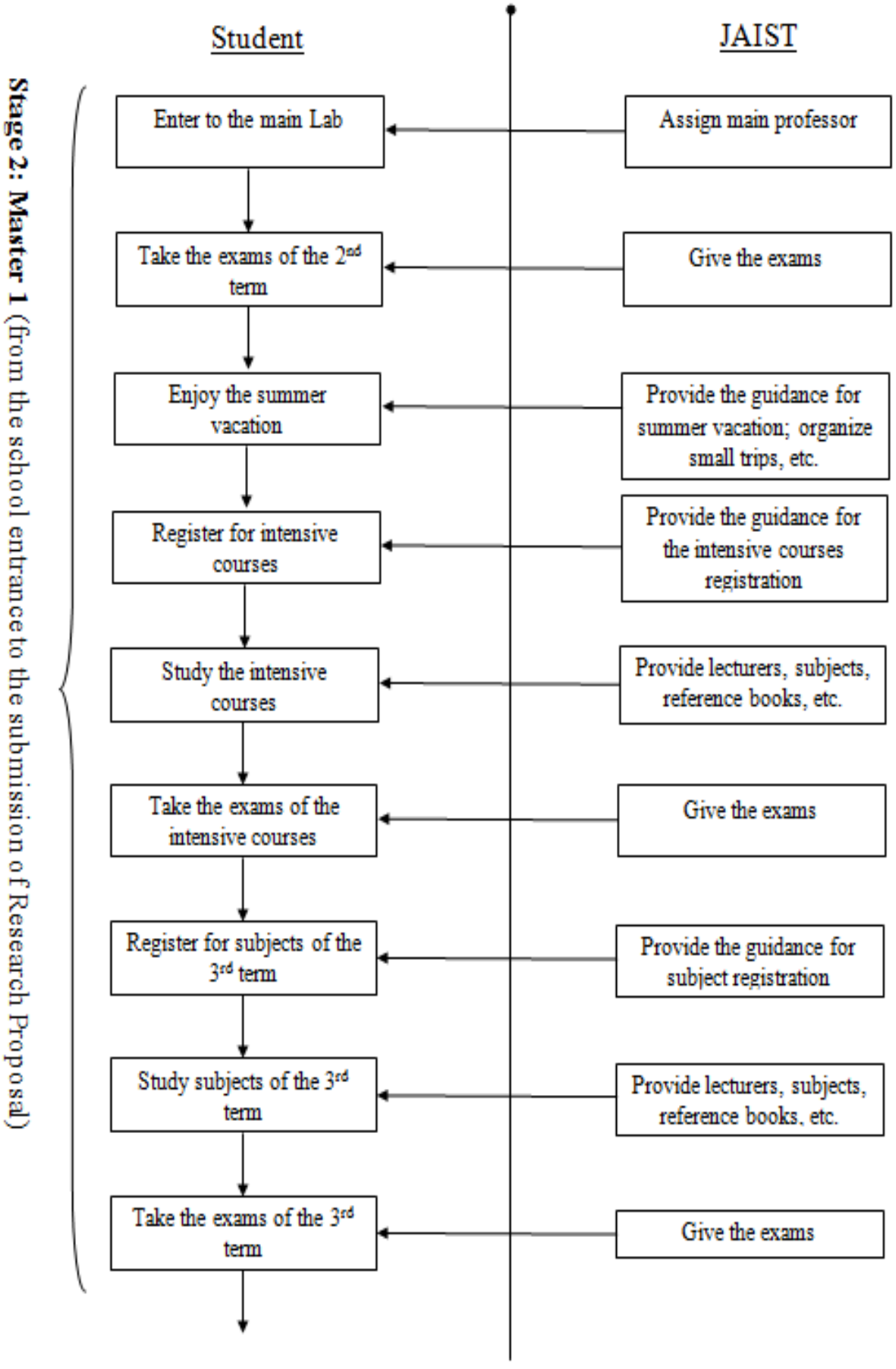
This scenario consists of three stages: Start, M1 and M2 stages.

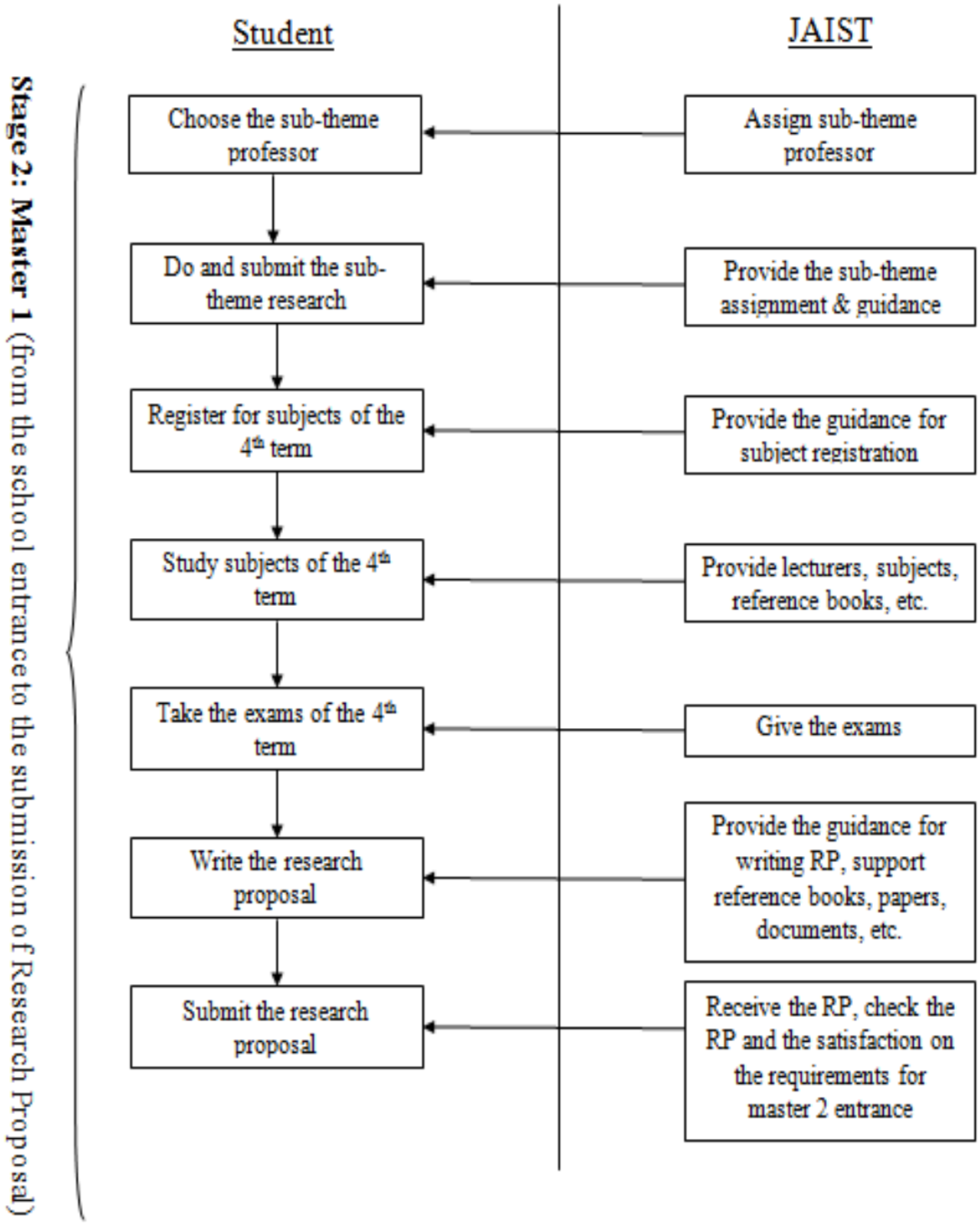
- Stage 1 (Start stage): from the entrance exam to the school entrance.
- Stage 2 (M1 stage): from the school entrance to the submission of Research Proposal.
- Stage 3 (M2 stage): from the Research Proposal to the graduation.

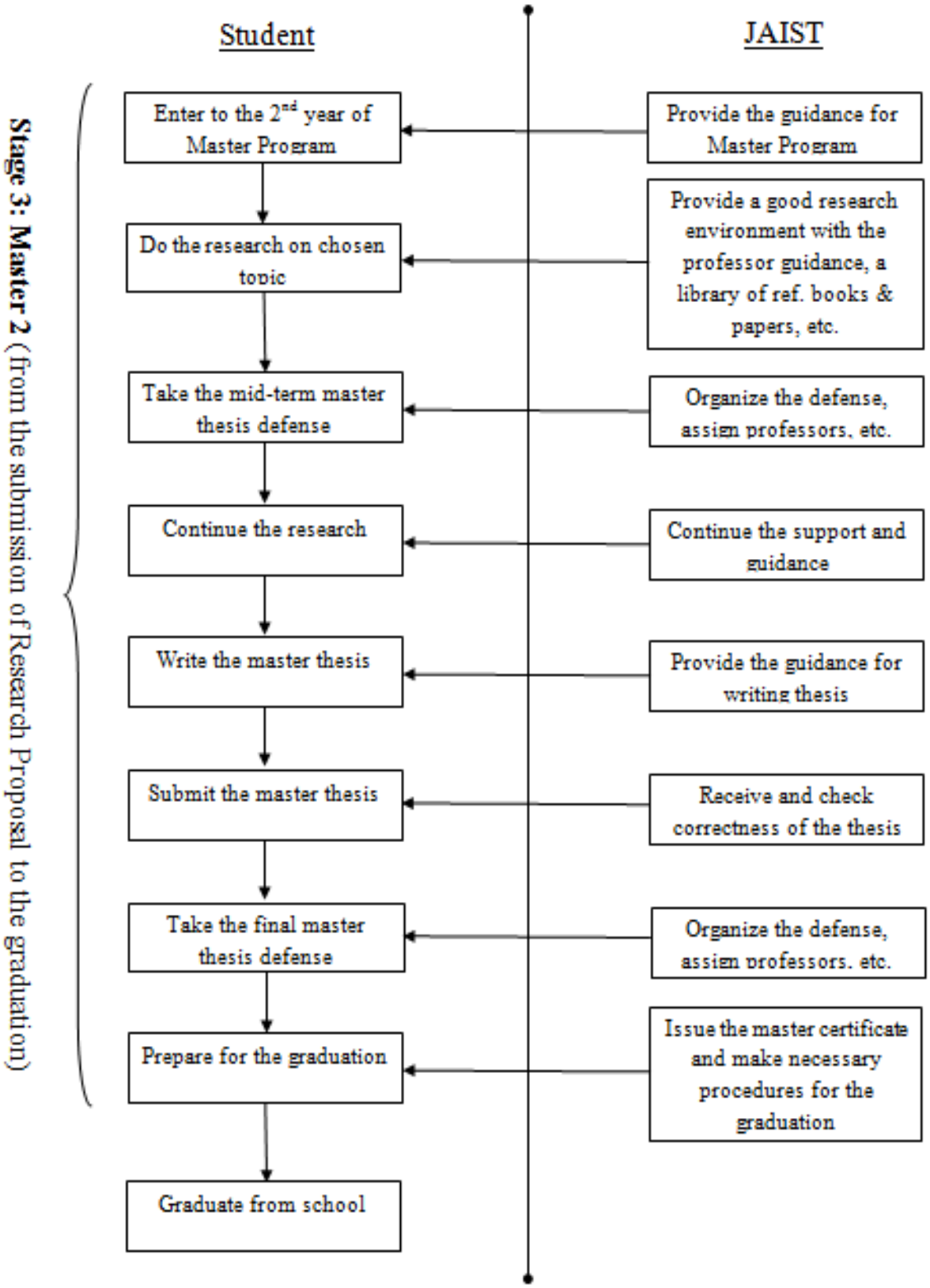
Figure 17: Scenario of M Program in JES.




Stage 2: Master 1 (from the school entrance to the submission of Research Proposal)
Student
JAIST
Enter to the temporary Lab
Assign temporary Lab and professor
Register for subjects of the 1st term
Provide the guidance for subject registration
Study subjects of the 1st term
Provide lecturers, subjects, reference books, etc.
Take the exams of the 1st term
Give the exams
Register for subjects of the 2nd term
Provide the guidance for subject registration
Study subjects of the 2nd term
Provide lecturers, subjects, reference books, etc.
Search for interested Lab
Introduction of research topic by professors
Take the interview 30 mins. by professor
Give details about research topic
Choose one of professors; register for desired Lab
Give guidance for Lab registration
Select the student by professor


Student
JAIST
Stage 2: Master 1 (from the school entrance to the submission of Research Proposal)
Enter to the main Lab
Assign main professor
Take the exams of the 2nd term
Give the exams
Enjoy the summer vacation
Provide the guidance for summer vacation; organize small trips, etc.
Register for intensive courses
Provide the guidance for the intensive courses registration
Study the intensive courses
Provide lecturers, subjects, reference books, etc.
Take the exams of the intensive courses
Give the exams
Register for subjects of the 3rd term
Provide the guidance for subject registration
Study subjects of the 3rd term
Provide lecturers, subjects, reference books, etc.
Take the exams of the 3rd term
Give the exams


Stage 2: Master 1 (from the school entrance to the submission of Research Proposal)
Student
JAIST
Choose the sub-theme professor
Assign sub-theme professor
Do and submit the sub-theme research
Provide the sub-theme assignment & guidance
Register for subjects of the 4th term
Provide the guidance for subject registration
Study subjects of the 4th term
Provide lecturers, subjects, reference books, etc.
Take the exams of the 4th term
Give the exams
Write the research proposal
Provide the guidance for writing RP, support reference books, papers, documents, etc.
Submit the research proposal
Receive the RP, check the RP and the satisfaction on the requirements for master 2 entrance


Stage 3: Master 2 (from the submission of Research Proposal to the graduation)
Student
JAIST
Enter to the 2nd year of Master Program
Provide the guidance for Master Program
Do the research on chosen topic
Provide a good research environment with the professor guidance, a library of ref. books & papers, etc.
Take the mid-term master thesis defense
Organize the defense, assign professors, etc.
Continue the research
Continue the support and guidance
Write the master thesis
Provide the guidance for writing thesis
Submit the master thesis
Receive and check correctness of the thesis
Take the final master thesis defense
Organize the defense, assign professors, etc.
Prepare for the graduation
Issue the master certificate and make necessary procedures for the graduation
Graduate from school

By using Scenarios, the students can know clearly about the main procedures they will do during the study of the program as well as the support from the School.

4.4 Business Process Modeling

In order to understand in more detail about the interaction between Student and School and how Student access services provided by School, we represent Student and School processes by using BPMN technology to creates the system workflows. The figures below show some example of created BPMs in JES.

We are now illustrating the following part of the general model of ES mentioned above:

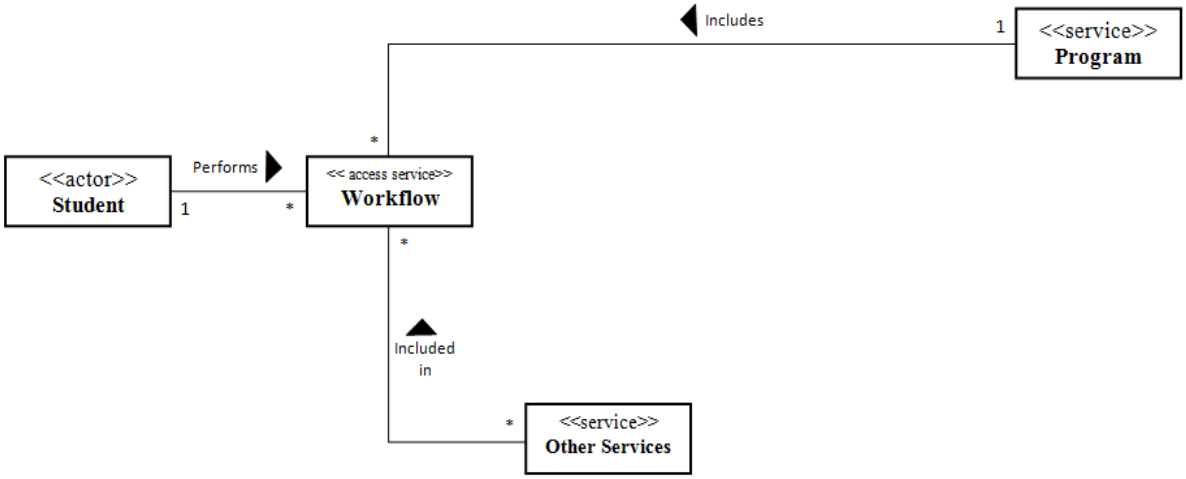


Figure 18: Program service part is the general model of ES.

And here is BPM for Program service:

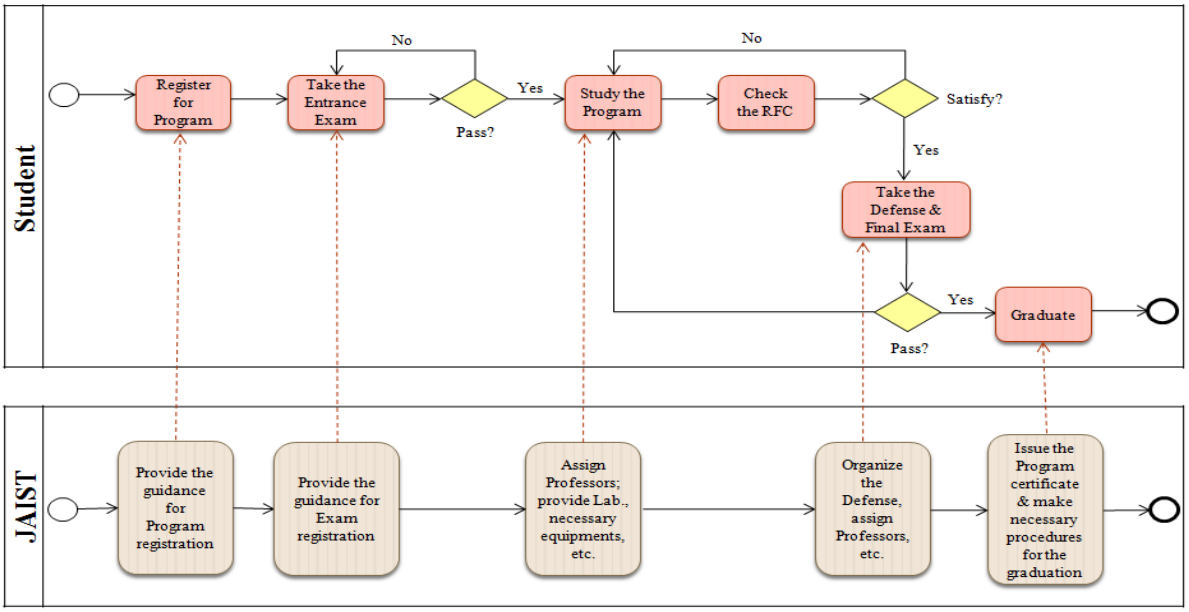


Figure 19: Example of BPM for Program service.

The model consists of two lanes: Student and JAIST lanes. In the Student lane, there are some processes performed by Student and the condition checking for some processes.

The goal of this research is to support for the students, so the building of BPMs will help the students to easily understand about the general and detailed processes they will perform during the study time in School, and then be able to easily find and choose services which are suitable for them.

Next, we will reach the following part of the general model of ES mentioned above:

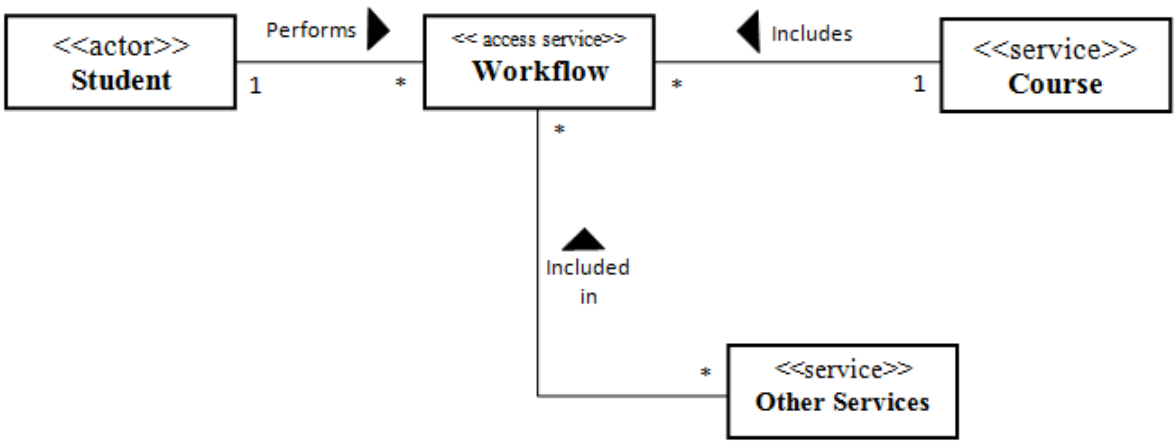


Figure 20: Course service part in the general model of ES.

And here is the BPM for Course service:

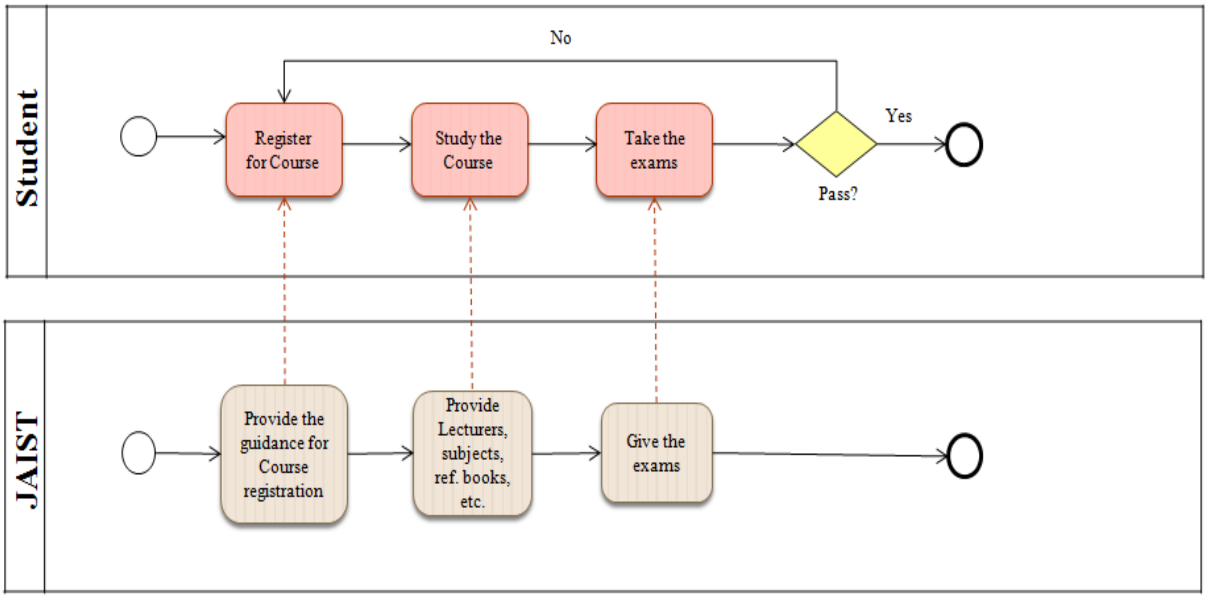


Figure 21: Example of BPM for Course service.

Then now, we are going to the following part of the general model of ES mentioned above:

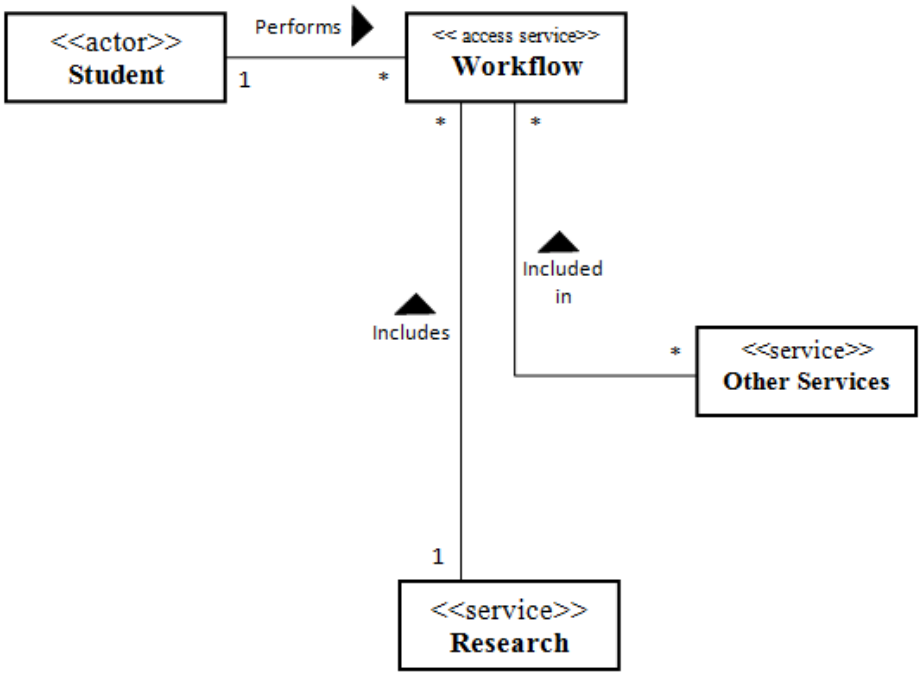


Figure 22: Research service part in the general model of ES.

And here is the BPM for Research service:

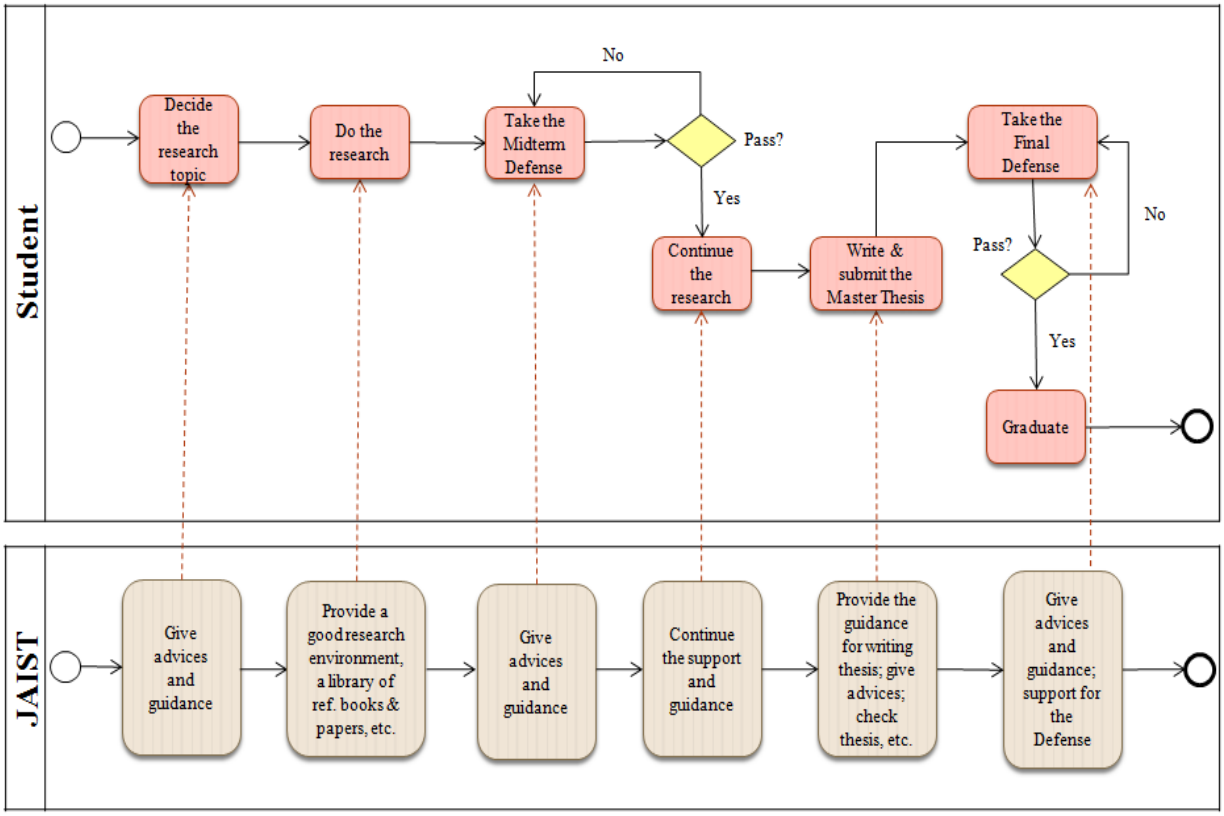


Figure 23: Example of BPM for Research service.

Because each instance of services has a different workflow from the others, so after having the general BPMs for services which provide the general view for processes, we draw the detailed BPM for each instance of services.

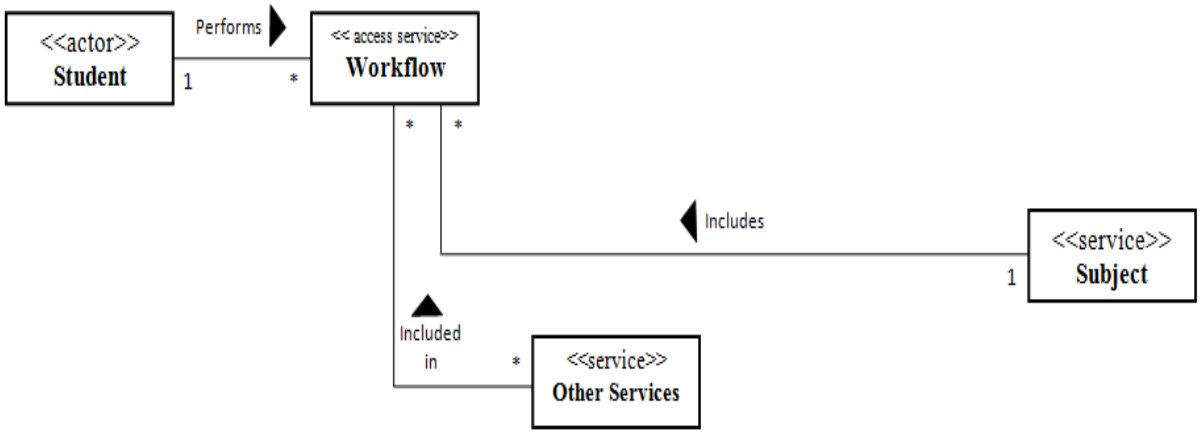


Figure 24: Subject service part in the general model of ES.

We are now illustrating the above part of the general model of ES mentioned earlier and here are some examples of created detailed BPMs of JES for Subject service:

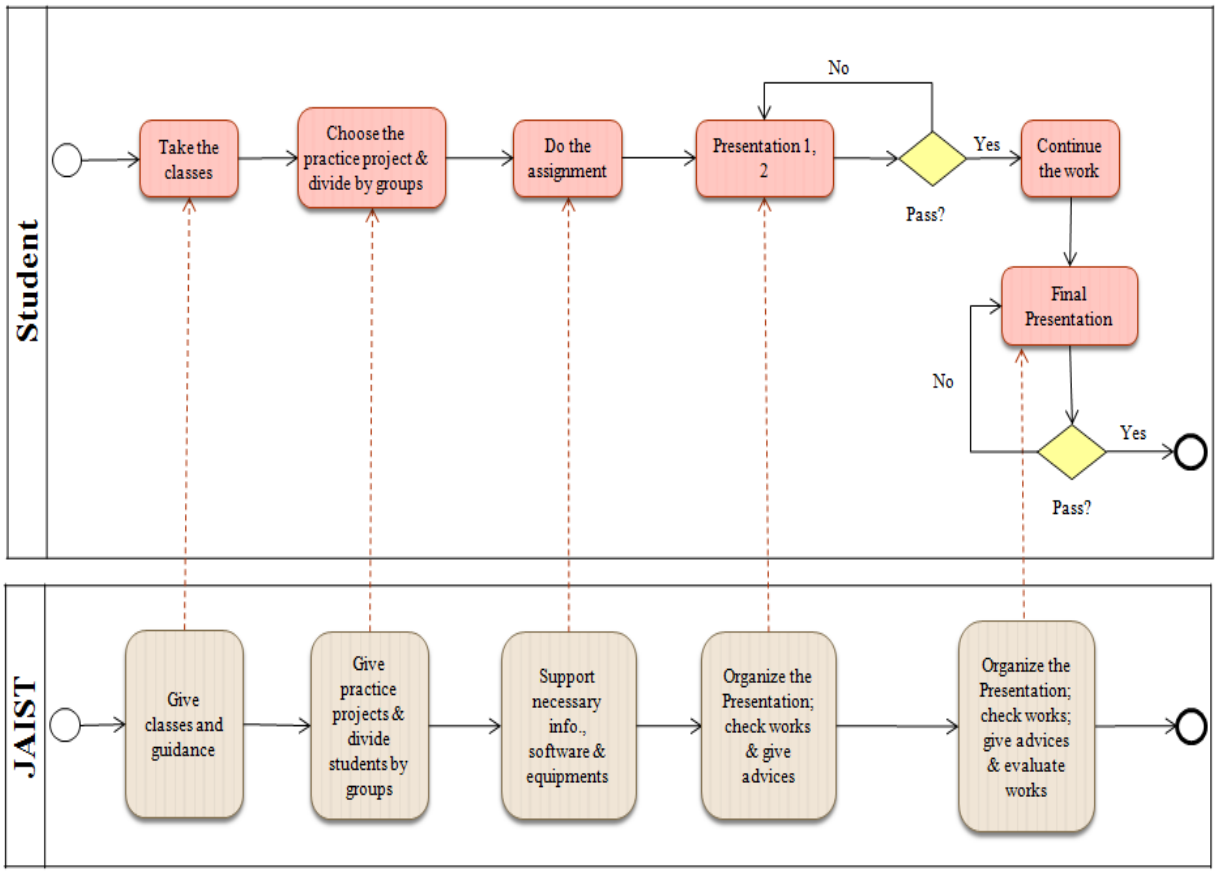


Figure 25: Example of BPM for [451] Subject service.

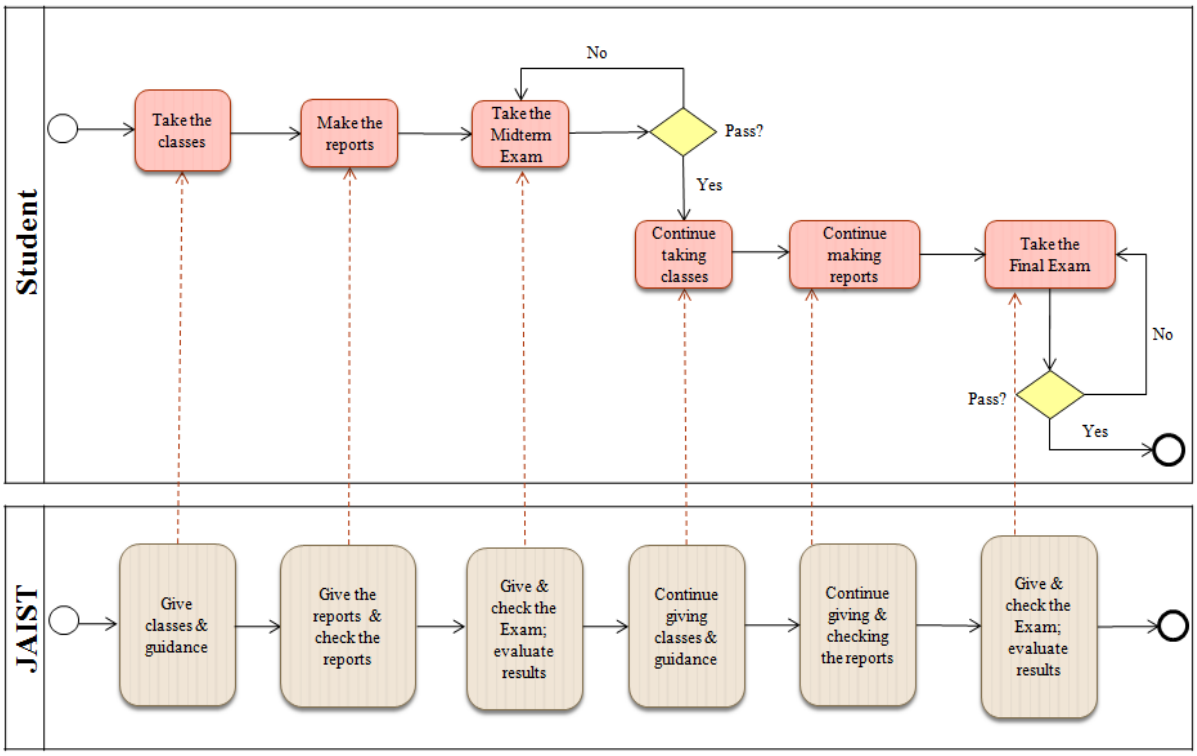


Figure 26: Example of BPM for [226] Subject service.

Chapter 5 - Aspects

5.1 Type of Aspects

As mentioned above, in AOSD, Aspects are the modularized concerns that cut across other concerns, or in other words, Aspects are the modularized crosscutting concerns that crosscut across several services. After making some analyses, we find out that in our research, there are three types of Aspects:

- Constraint Aspects
- Relationship Aspects
- Extension Aspects

5.2 Constraint Aspects

In the general model of ES, the constraint is included in the following part and we are going to discuss about it now:

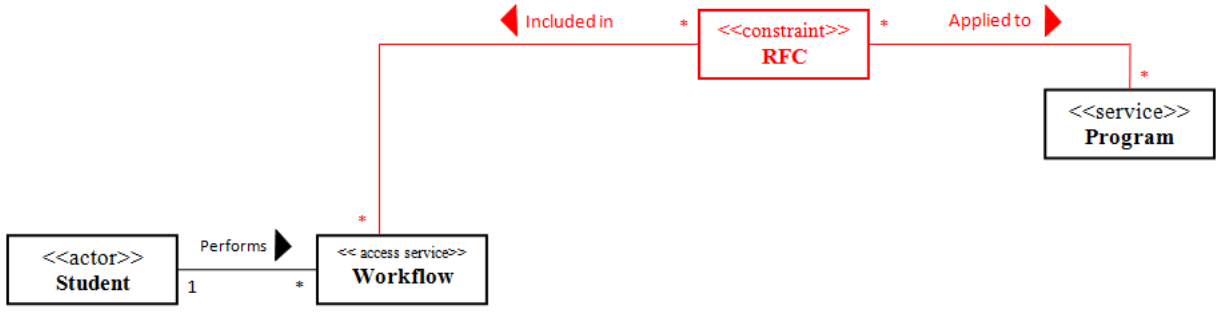


Figure 27: Constraint part in the general model of ES.

5.2.1 Definition of Constraints

A constraint is a condition that must be satisfied to process further. Example of constraints includes: business rules, technical requirements, etc. In this research, constraints are used to specify rules or requirements that are applied to the services.

5.2.2 Example of Constraints

In order to complete a program in School, the students must follow some rules or requirements for completion (RFC) the program. Below is an example of course requirements for completion of Master program in JES. To complete it, students must choose whether they will write Master Thesis or Project Report and both of them have different RFCs.

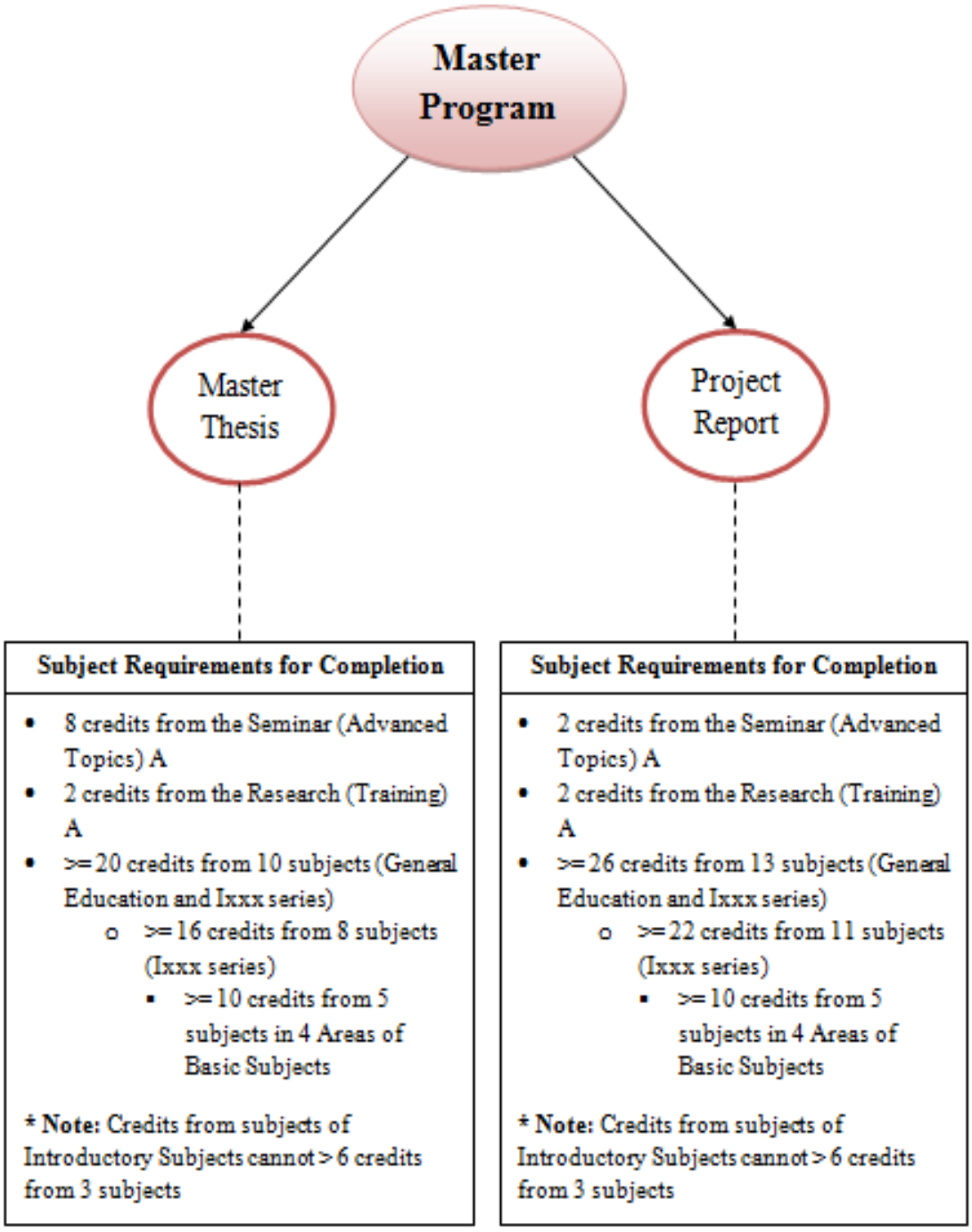


Figure 28: Example of Constraints in JES.

Because constraints are rules or requirements that are applied to the services, they can also be represented in BPMs. Here is an example of RFC from the BPM for M2 program mentioned above:

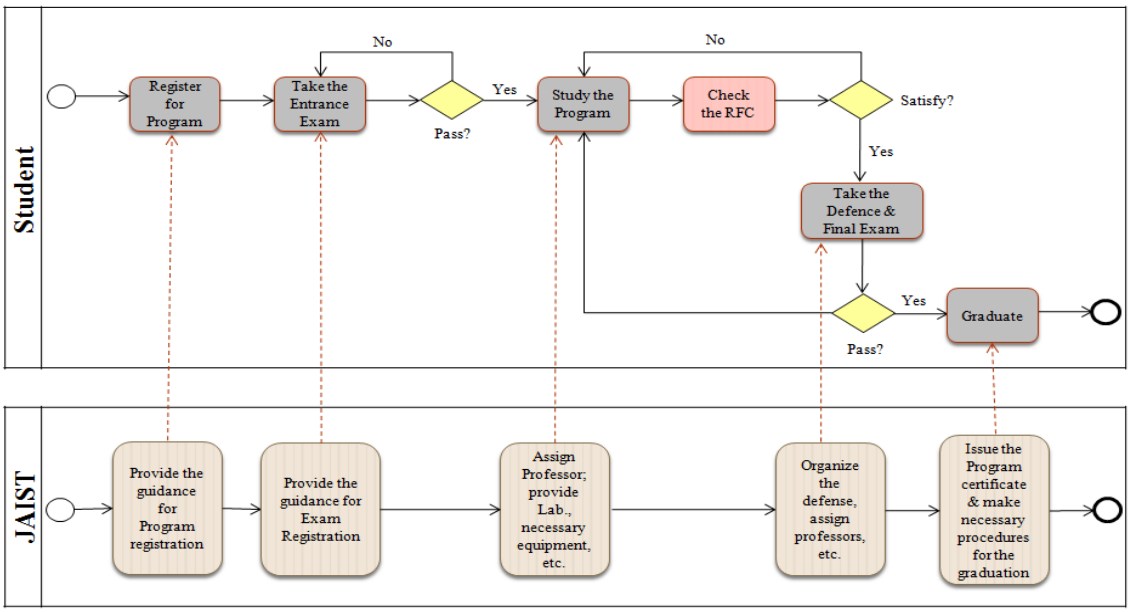


Figure 29: Example of Constraints in BPM.

5.2.3 Definition of Constraint Aspects

On the other hand, the same RFC can be applied for different services and they may crosscut those services as shown in this figure:

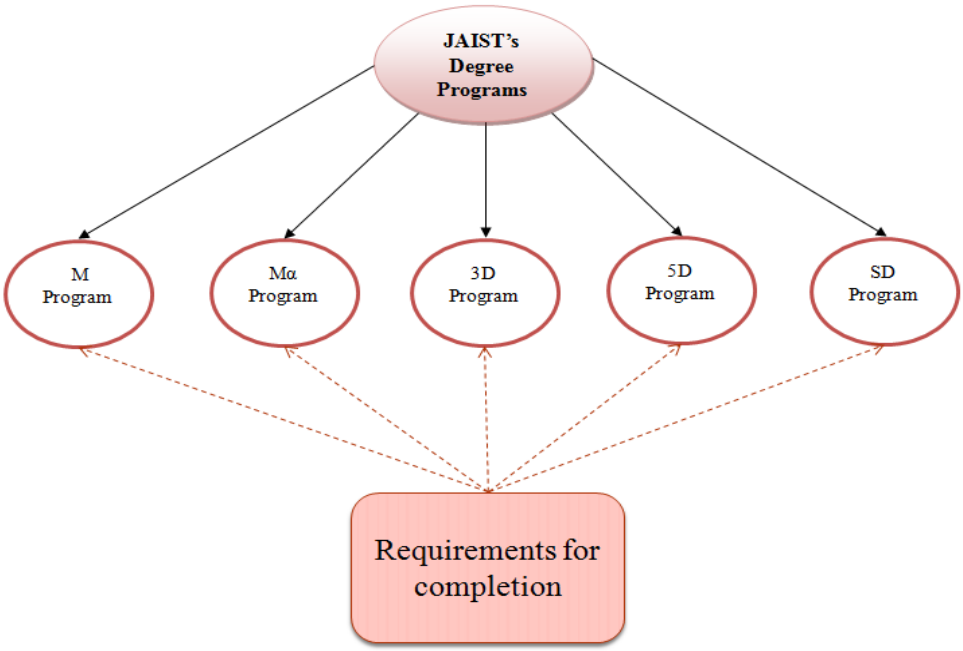


Figure 30: Example of Constraints in BPM.

Therefore, we can treat them as Aspects and called them Constraint Aspects.

5.2.4 Structure of Constraint Aspects

The figure below shows the structure of a Constraint Aspect:

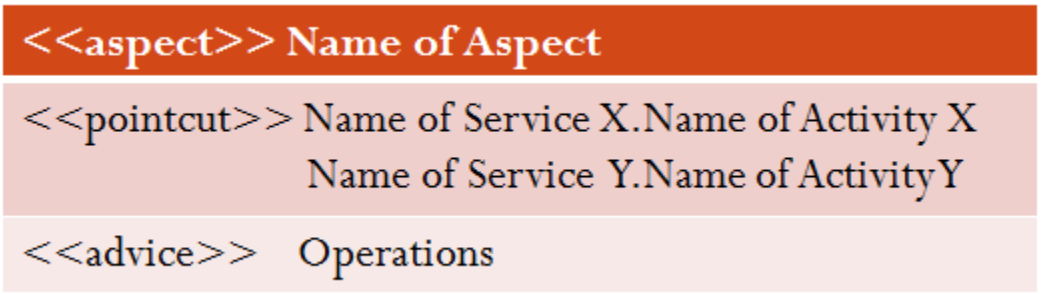


Figure 31: Structure of CA.

Back to the example of the previous subsection, we also have the structure of Aspect named "CheckRFC" as follows:

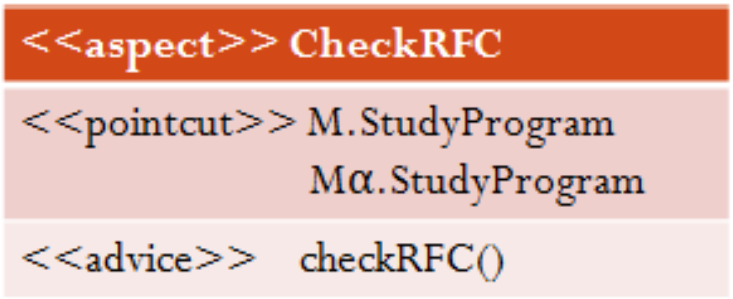


Figure 32: Structure of "CheckRFC" Aspect.

To apply the concept of Constraint Aspects in JES to solve the problem of constraints that crosscut across several services, we separate those constraints from the main workflow and processes them independently so that they can be easily updated, changed or removed.

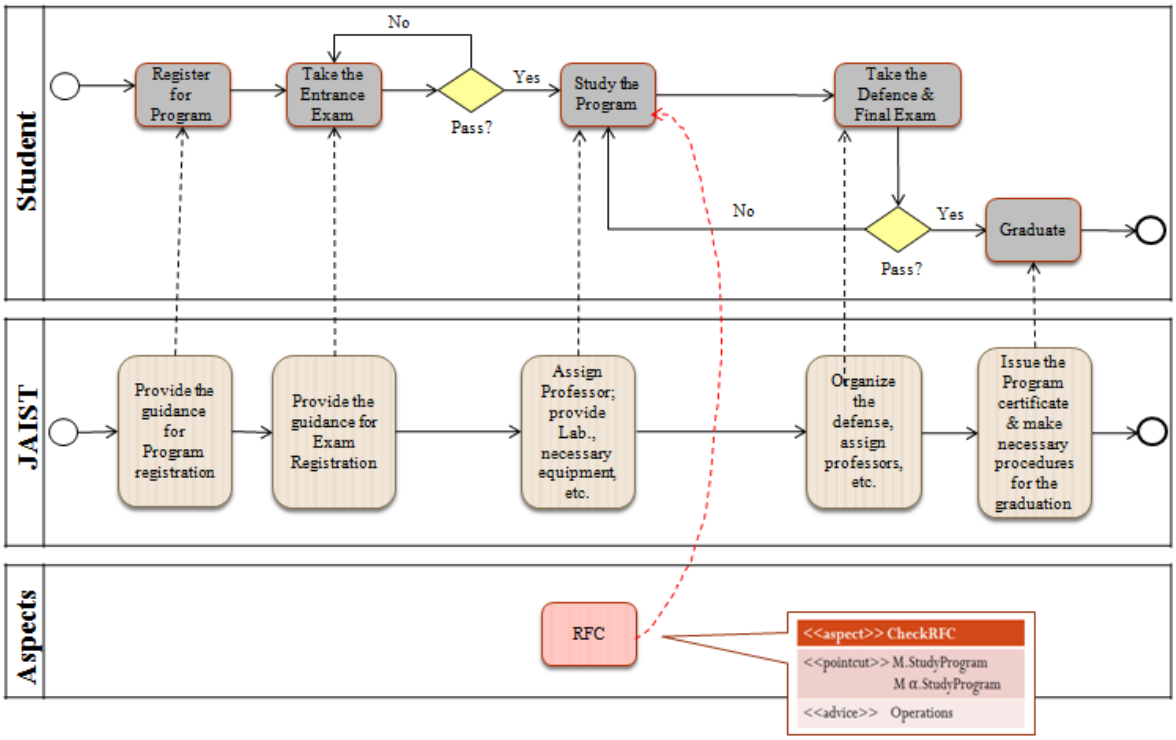


Figure 33: Example of Constraint Aspects in BPM.

Consider the table below which lists the Program services in JES and their RFCs:

Table 1: Table of Programs and RFCs in JES.

Program Name	**Requirements for Completion**	

M Program	Master Thesis	• 8 credits from Seminar (Advanced Topics) A
		• 2 credits from Research (Training) A
		• >= 20 credits from 10 subjects (General Education and Ixxx series)
		  ○ >= 16 credits from 8 subjects (Ixxx series)
		  ▪ >= 10 credits from 5 subjects in 4 Areas of Basic Subjects
		*** Note:** Credits from subjects of Introductory Subjects cannot > 6 credits from 3subjects.

	Project Report	• 2 credits from Seminar (Advanced Topics) A
		• 2 credits from Research (Training) A
		• >= 26 credits from 13 subjects (General Education and Ixxx series)
		  ○ >= 22 credits from 11 subjects (Ixxx series)
		  ▪ >= 10 credits Subjects from 5 subjects in 4 Areas of Basic Lectures
		*** Note:** Credits from subjects of Introductory Subjects cannot > 6 credits from 3 subjects.

Mα Program	Same as in M Program	
3D Program		• 6 credits from Seminar (Advanced Topics) B • 4 credits from Research (Training) B • >= 10 credits from 5 subjects (Basic, Intermediate, Advanced Subjects and Seminars) ○ >= 10 credits from 5 subjects in 3 Areas (Basic, Intermediate, Advanced Subjects and Seminars) *** Note:** If the credits earned in the M Program exceed the RFC for M Program, then the excess <= 8 credits can be transferred in 3D Program.

<table>
<tr><td rowspan="2">5D Program</td><td>Master Thesis</td><td>

8 credits from Seminar (Advanced Topics) A
2 credits from Research (Training) A
6 credits from Seminar (Advanced Topics) B
4 credits from Research (Training) B
>= 20 credits from 10 subjects (General Education and Ixxx series)

>= 16 credits from 8 subjects (Ixxx series)

>= 10 credits from 5 subjects in 4 Areas of Basic Lectures

* Note: Credits from subjects of Introductory Subjects cannot > 6 credits from 3 subjects.

>= 22 credits from 11 subjects (General Education and Ixxx series)

>= 10 credits from 5 subjects in 4 Areas of Basic Subjects (in M Program)
>= 10 credits from 5 subjects in 3 Areas of Basic, Intermediate, Advanced Subjects and Seminars (in D Program)

* Note: If the credits earned in the M Program exceed the RFC for M Program, then the excess <= 8 credits can be transferred, but still need to obtain 2 credits in D Program.
</td></tr>
<tr><td>Project Report</td><td>

2 credits from Seminar (Advanced Topics) A
2 credits from Research (Training) A
6 credits from Seminar (Advanced Topics) B
4 credits from Research (Training) B
>= 26 credits from 13 subjects (General Education and Ixxx series)

>= 22 credits from 11 subjects (Ixxx series)

>= 10 credits from 5 subjects in 4 Areas of Basic Subjects

</td></tr>
</table>

		*** Note:** Credits from subjects of Introductory Subjects cannot > 6 credits from 3 subjects. • >= 28 credits from 14 subjects (General Education and Ixxx series) o >= 10 credits from 5 subjects in 4 Areas of Basic Subjects (in M Program) o >= 10 credits from 5 subjects in 3 Areas of Basic, Intermediate, Advanced Subjects and Seminars (in D Program) *** Note:** If the credits earned in the M Program exceed the RFC for M Program, then the excess <= 8 credits can be transferred, but still need to obtain 2 credits in D Program.

From the table above, we can see that:

- RFC of M Program is the same as Mα Program and is alos included in D5 Program.
- RFC of D3 Program is included in D5 Program.

Therefore, we define them as two Aspects named "CheckMRFC" and "CheckDRFC" with the structures as follows:

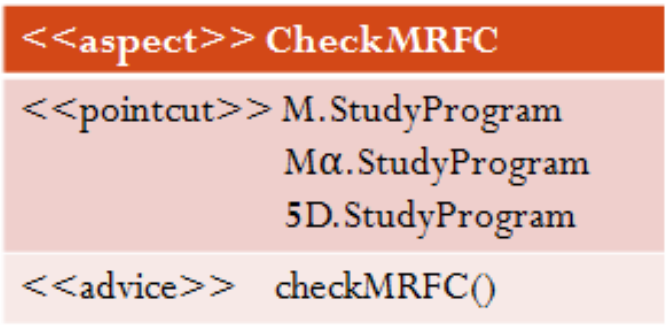


Figure 34: Structure of "CheckMRFC" Aspect.

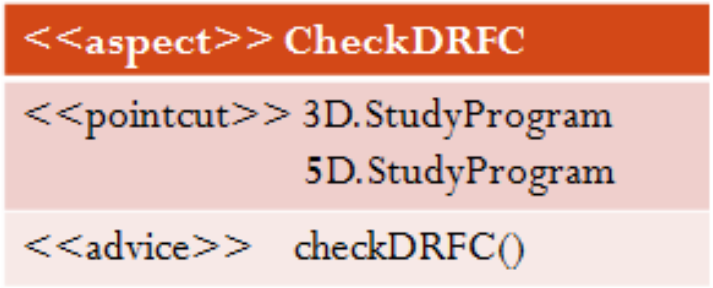


Figure 35: Structure of "CheckDRFC" Aspect.

5.3 Relationship Aspects

No object stands alone – without relationship, most objects are meaningless. Therefore, relationships between objects in OOP are as important as the objects themselves [6]. In the general model of JES, the relationship between subjects is included in the following part and we are going to discuss about it now:

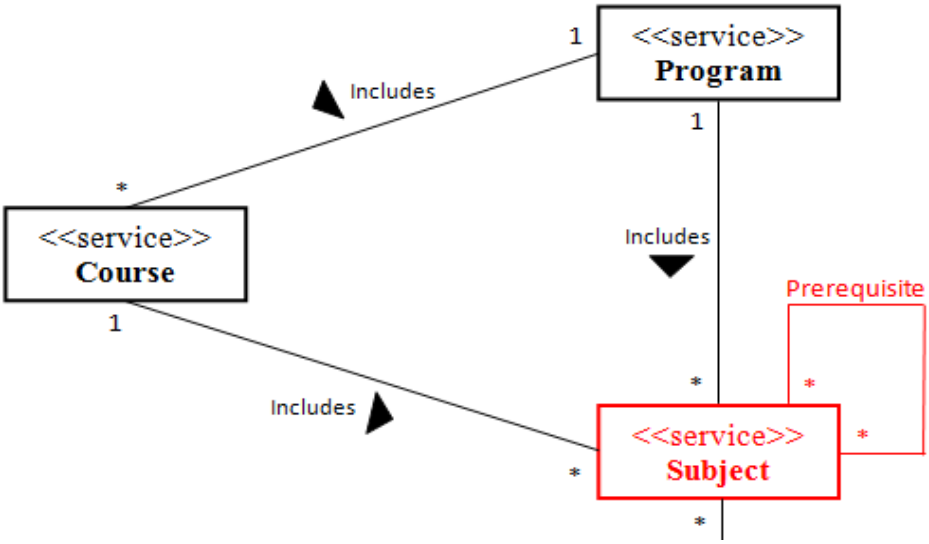


Figure 36: Relationship part in the general model of ES.

5.3.1 Problem in ESs

There are many subjects provided by School ES, but they have quite complex relationships with each other. For example, in JES, if you want to take the subject I212, you must take the subject I114 first or if you already took the subject I616, then you cannot register for the subject I212.

The Table 1 shows an example of a list of subjects and their prerequisites in JES.

Table 2: Table of Subjects and Prerequisites in JES.

Type of Subjects	Subject Code	Subject Name	Prerequisite
Introductory Subjects	I111	Algorithms & Data Structures	None
	I112	Computer Systems	None
	I114	Fundamental Mathematics for Information Science	None
	I115	Digital Logic and Computer Design	None

	I116	Programming Laboratory I	None
	I117	Programming Laboratory II	None
	I118	Graphs and Automata	Had: I113 or I216 -> cannot register
	I119	Statistics in Information Science	None
Basic Subjects	I211	Mathematical Logic	None
	I212	Analysis for Information Science	Need: I114
	I213	Discrete signal processing	None
	I214	System Optimization	None
	I215	Artificial Intelligence	None
	I216	Computational Complexity and Discrete Mathematics	Need: I118 Had: I222 -> cannot register
	I217	Functional Programming	None
	I218	Computer Architecture	None
	I219	Software Design Methodology	None
	I223	Natural Language Processing I	None
	I225	Statistical Signal Processing	Need: I114 and I119
	I226	Computer Networks	None
	I232	Information Theory	Had: I436 -> cannot register
	I233	Operating Systems	Had: I426 -> cannot register
	I234	Foundation of Software Environment	Had: I425 -> cannot register

Intermediate Subjects	I411	Pattern Analysis and Recognition	None
	I413	Theoretical Computer Science/ Lambda-Calculus	None
	I414	Natural Language Processing II	Need: I223
	I416	Parallel Processing	None
	I419	Image Information Science	Need: I212 and I213
	I427	System Control Theory	None
	I429	Intelligent Agents	Need: I215
	I431	Theory of Algorithms	Need: I111 and I216
	I432	Theory of Discrete-State Systems	Need: I211 and I216
	I433	Numerical Computing	Need: I114
	I435	Software Architecture	Need: I219
	I437	Coding Theory	Need: I232
	I438	Exercises on Graph Theory	Need: I118
	I439	Speech Signal Processing	Need: I212, I213 and I225
	I440	Enhanced Operating System	Need: I233
	I441	Advanced Computer Networks	Had: I428 -> cannot register
	I442	Advanced System Software Laboratory	Need: I116 and I117 Had: I224 -> cannot register
	I443	Foundation of Software Verification	Need: I118 and I211

	I444	Embedded Software Engineering	Need: I219 Had: I641 -> cannot register
	I446	Computer Systems Performance Analysis	None
	I447	Database Systems	None
	I448	Distance Learning System	None
	I450	Network Design	Need: I226
	I451	Software Design Laboratory	Need: I219
	I473	Hardware/ Software Co-design	None
	I478	IT Project Management	None
	I470E	Integration Architecture	None
	I472E	Co-design	None
	I474E	Embedded Software Engineering	Need: I219
	I476E	Component Technology and Middleware	None
	I490E	Functional Programming and Formal Methods	None
	I482F	Advanced Algorithms	Need: I111 and I431
	I484F	Robotics	None
	I486F	Logic and Language	Need: I215 and I211
	I450G	Basics of Software Science	None
	I452G	Design Verification – Advanced	None
	I453G	Formal Specification (basic)	None
	I455G	Formal Specification (security)	None

	I457G	Software Patterns	None
	I460G	Real-time Model Checking	None
Advanced Subjects	I613	Formal Methods	Need: I111 and I118
	I616	Human Information Processing	Need: I212 and I213
	I631	Foundation of Computational Geometry	Need: I216 and I431
	I640	Automated deduction	Need: I118 and I211
	I646	Entertainment Informatics	None
	I648	Reasoning about Interaction and Regulation	None

According to this table, we can see that there are three types of Prerequisites in JES:

- None: means there in no relationship with other subjects.
- Needs: means there is a need of taking another subject before taking this subject.
- Cannot register: means that students cannot take this subject if they already took another one.

The relationship between subjects or the prerequisites can be seen very clearly in the following figure:

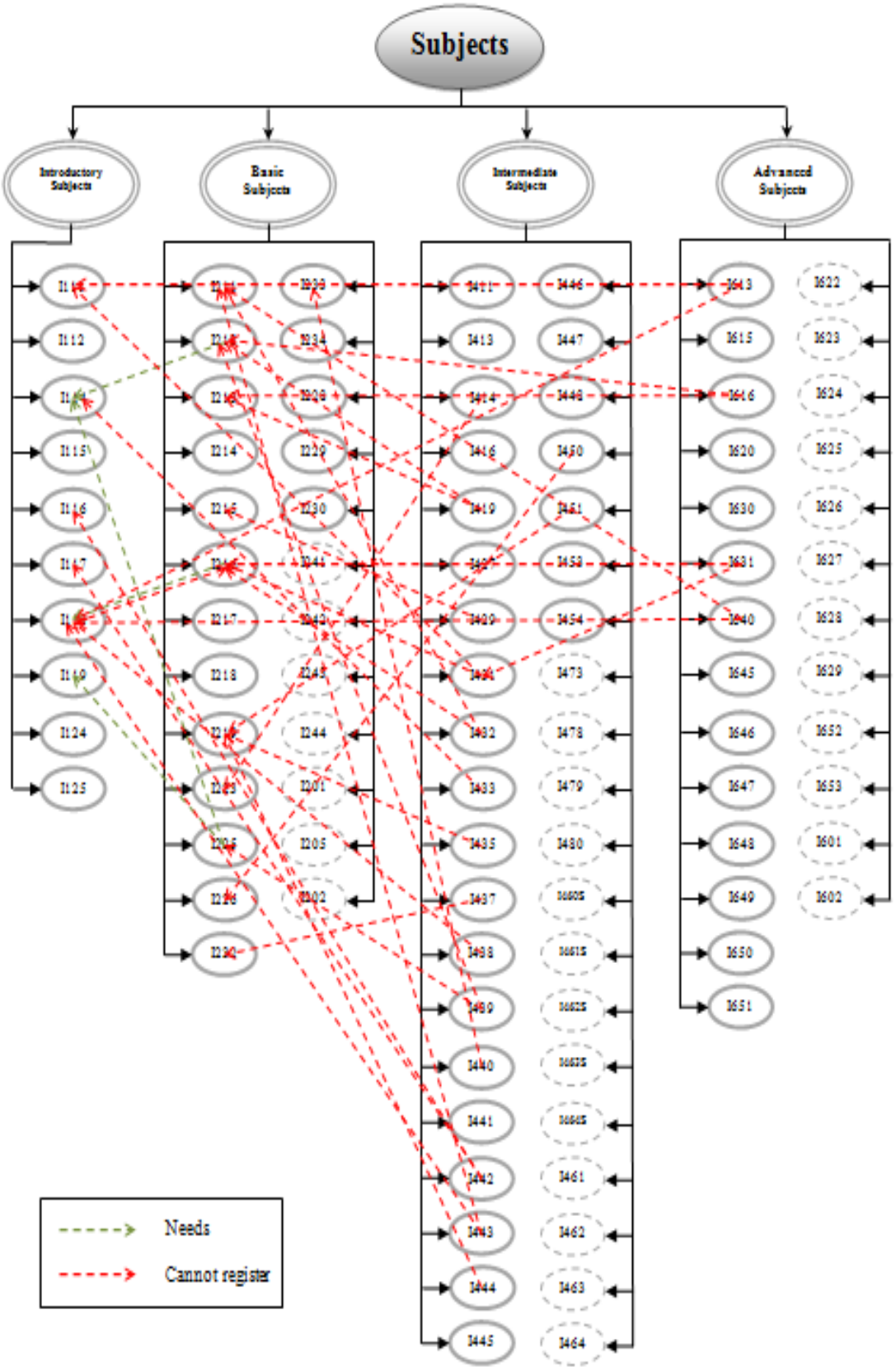


Figure 37: Relationship between Subjects in JES.

5.3.2 Definition of Relationship Aspects

Suppose we have three subjects: S1, S2 and S3. In order to register for S2, we must obtain credits from S1 first, so the S1 is needed by S2, or S2 "needs" S1. On the other hand, if we already took the S3, then we cannot register for the S2.

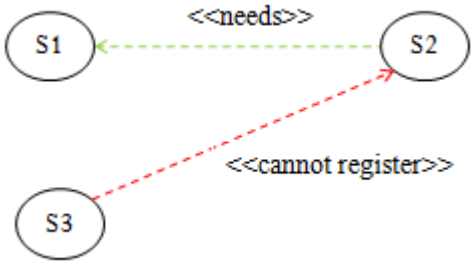


Figure 38: S1 and S3 have relationships with S2.

To deal with these relationships, usually we will store their information in both S1, S2 and S3 as shown in this figure:

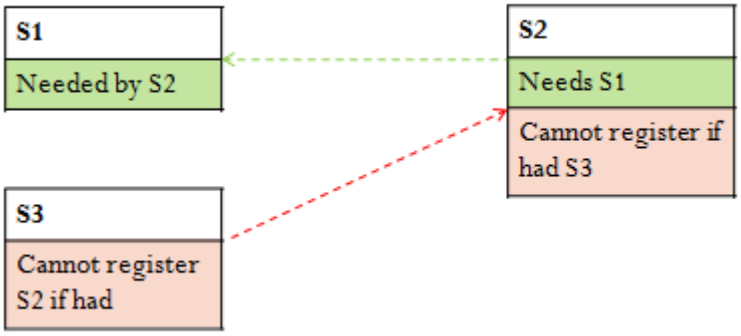


Figure 39: Information about relationships is stored in each subject.

In the situation where there are just 3 or 10 subjects in the system, it may work normally. But imagine where there are 100 or more subjects, it will be very hard and redundant when all subjects must store the information about their relationships with the others.

Moreover, most of OOP languages provide little support for such relationships. Relationships are typically hard-coded into the participating classes result in tangled code and classes become harder to understand and cannot be reused independently [6].

We can see that a relationship is shared in at least 2 objects or in other words, it crosscuts these objects. Therefore, we can treat them as separate concerns that cross-cut their participants and model them explicitly as Aspects, called Relationship Aspects (RA). Aspects keep relationships independent of their participants, making the resulting programs easier to read, write and reuse. The

implementation of the relationship can be improved, removed, or replaced, without affecting the participating classes.

5.3.3 Structure of Relationship Aspects.

The figure below shows the structure of an Relationship Aspect:

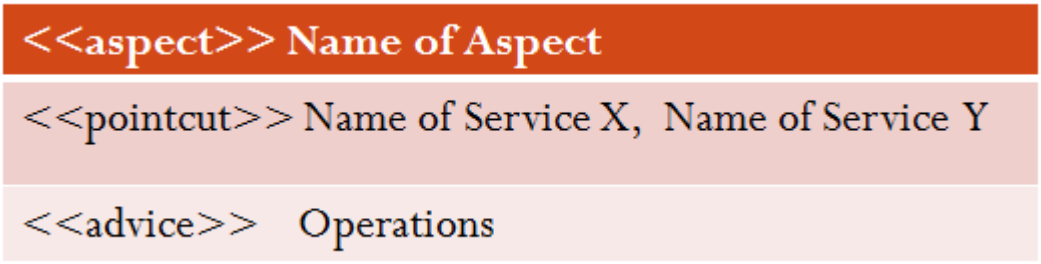


Figure 40: Structure of RA.

Back to the example of the previous subsection, we also have the structure of Aspect named "Needs" and Aspect named "CannotRegister" as shown in the Figure 32:

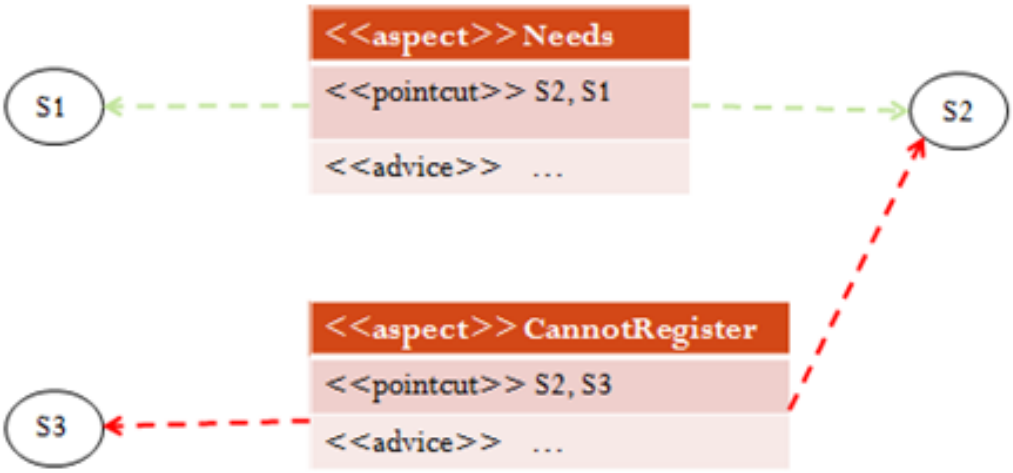


Figure 41: Example of Relationship Aspects & their structure.

Where:

- "S2, S1" means S2 needs S1
- "S2, S3" means cannot register for S2 if had S3

To apply the concept of Relationship Aspects in JES to solve the problem of complex relationship between subjects, we made a table which includes a list of Aspect "Needs", Aspect "CannotRegister" and their poincuts as follows:

Table 3: Example of Relationship Aspects in JES.

Aspect Name	Pointcut
<<aspect>> Needs	I212, I114
	I216, I118
	I225, I114
	I225, I119
	I414, I223
	I419, I212
	I419, I213
	I429, I215
	I431, I111
	I431, I216
	I432, I211
	I432, I216
	I433, I114
	I435, I219
	I437, I232
	I438, I118
	I439, I212
	I439, I213
	I439, I225
	I440, I233
	I442, I116
	I442, I117
	I443, I118
	I443, I211
	I444, I219
	I450, I226
	I451, I219
	I474E, I219
	I482F, I111
	I482F, I431
	I486F, I215
	I486F, I211
	I613, I111
	I613, I118
	I616, I212
	I616, I213

Aspect Name	Pointcut
<<aspect>> CannotRegister	I118, I113
	I118, I216
	I216, I222
	I232, I436
	I233, I426
	I234, I425
	I441, I428
	I442, I224
	I444, I641

	I631, I216
	I631, I431
	I640, I118
	I640, I211

5.4 Extension Aspects

5.4.1 Definition of Extension Aspects

As mentioned above, in AOSD, Aspects are the modularized concerns that cut across other concerns or in other words, Aspects are the modularized crosscutting concerns that crosscut across several functions.

Basically, use cases refer to the functions performed by the system. In the relation to AOSD, a use case corresponds most directly to a concern. It is a crosscutting concern because it crosscuts the components that realize it. Therefore, Jacobson gave a new concept about the Aspects which are the Use case extensions [7].

Based on the basic definition and Jacobson's concept about Aspects and after making some analyses, we came out with the new definition for the other Aspects type in our research. Aspects can refer to the services provided by the School and Teacher actors if those services are the extensions of main services provided by School and Teacher. We called them as Extension Aspects (EA).

5.4.2 Example of Extension Aspects

Suppose we have some Teacher services that crosscut School services in JES as illustrated in the figure below:

Teacher / JAIST	Service T1	Service T2	Service T3	Service T4
Program	✓		✓	
Course		✓		✓
Service J1	✓	✓	✓	
Service J2	✓			✓

Figure 42: JAIST and Teacher Services cut crossing.

From the previous table, we have a diagram (in Figure 43) which represents more clearly about the Teacher services that crosscut JAIST services and these services can be the candidates for the Extension Aspects.

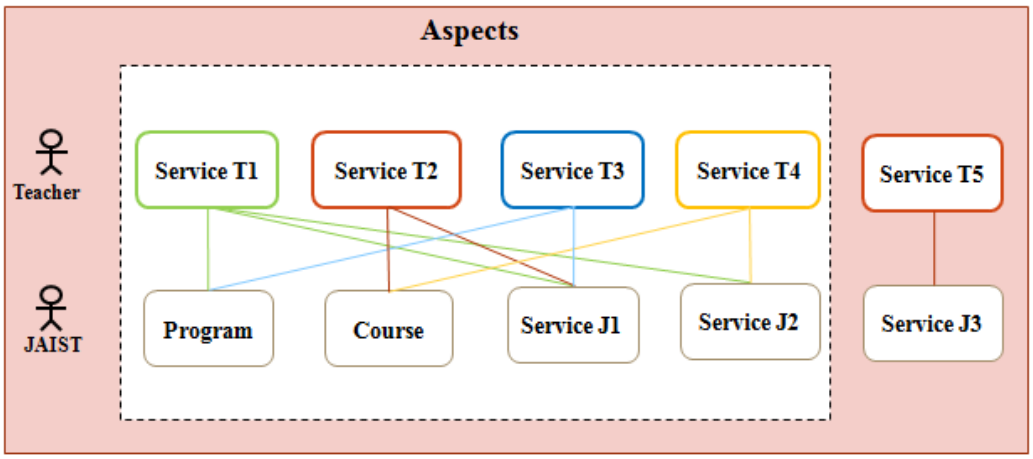


Figure 43: Example of Teacher's Services crosscut JAIST's Services.

5.4.3 Structure of Extension Aspects.

The figure below shows the structure of an Extension Aspect:

<<aspect>> Name of Aspect
<<pointcut>> Name of Service X Name of Service Y
<<advice>> Operations

Figure 44: Structure of EA.

Back to the example of the previous subsection, we also have the structure of Aspect named "ProvideServiceT1", Aspect named "ProvideServiceT2", Aspect named "ProvideServiceT3" and Aspect named "ProvideServiceT4" as shown in the Figure 37:

<<aspect>> ProvideServiceT1
<<pointcut>> ProvideProgram ProvideCourse ProvideServiceJ1
<<advice>> Operations

<<aspect>> ProvideServiceT2
<<pointcut>> ProvideCourse ProvideServiceJ1
<<advice>> Operations

<<aspect>> ProvideServiceT3
<<pointcut>> ProvideProgram ProvideServiceJ1
<<advice>> Operations

<<aspect>> ProvideServiceT4
<<pointcut>> ProvideCourse ProvideServiceJ2
<<advice>> Operations

Figure 45: Structure of example EA in JES.

The previous example can also be represented in UML diagram as shown below:

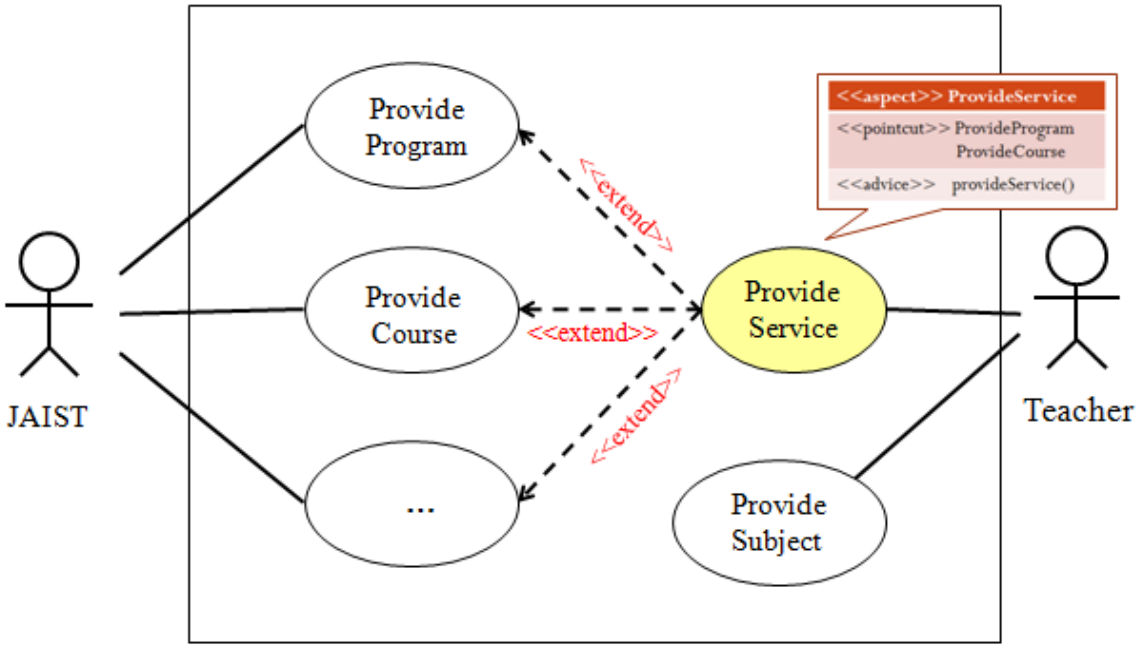


Figure 46: Example of EA in UML.

5.5 General model of educational system with Aspects

There are many ways to model a system. By using Service-oriented Modeling and Aspect-oriented Technology, we create our general model of ES.

The general model of ES consists of three actors: Student, School and Teacher. During the Study Line, student will perform many processes and each process is related to many services provided by School. Moreover, each services or instance of those services is related to the services provided by Teacher.

Besides, in the system, there are also crosscutting concerns (Aspects) that crosscut several services. They are: the relationships between School's services, the constraints that are applied to School's services as well as processes and the extension services that extend service provided by School. To build an completed model, we also include those Aspects in the model as illustrated below:

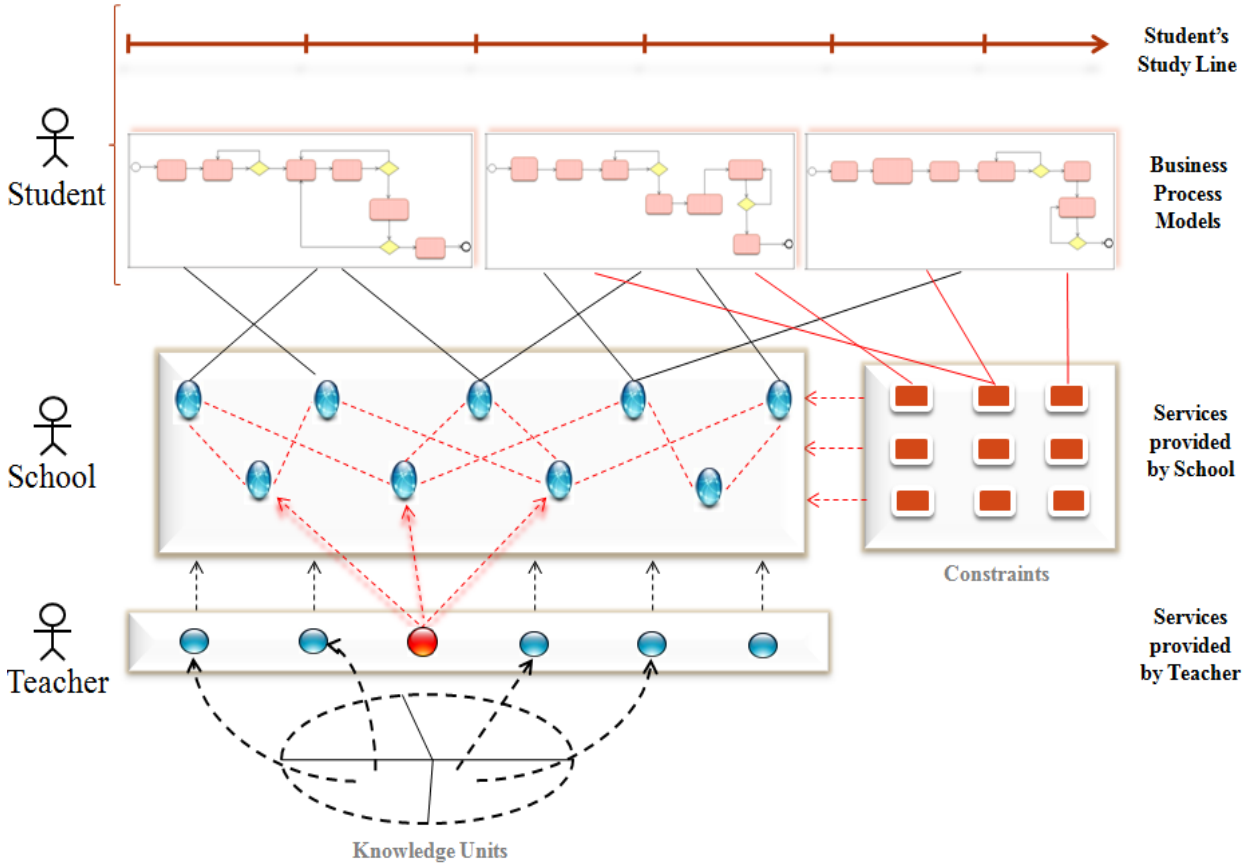


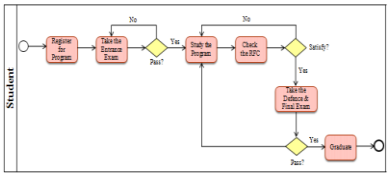
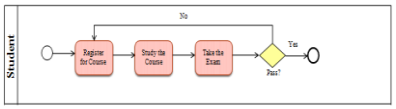
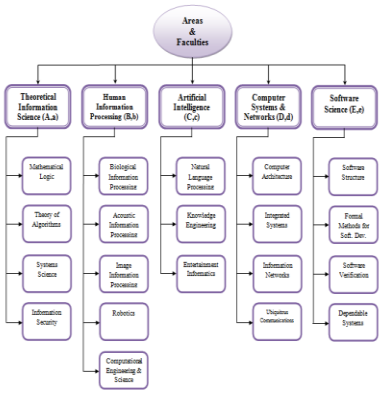
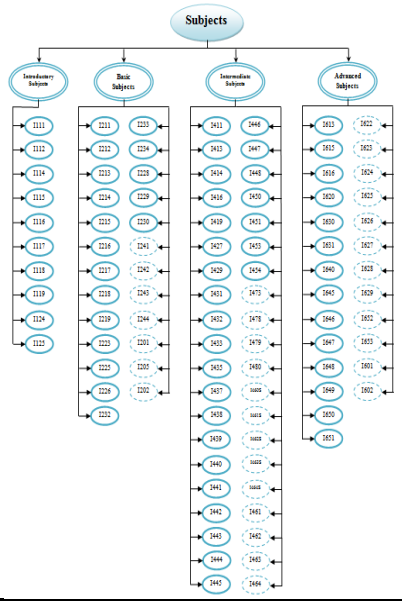
Figure 47: The general model of educational system (with Aspects).

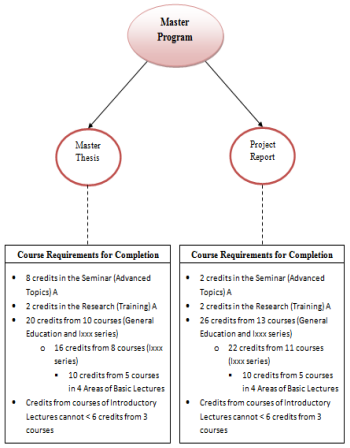
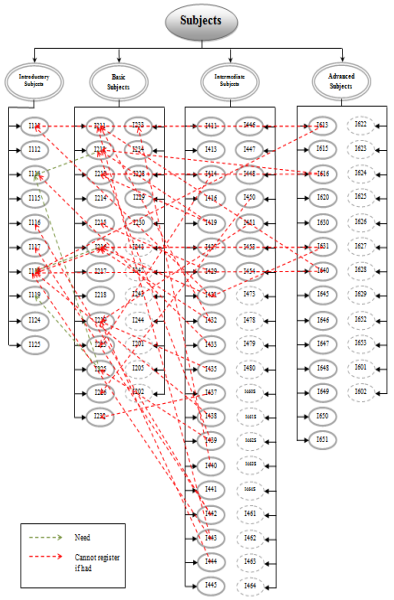
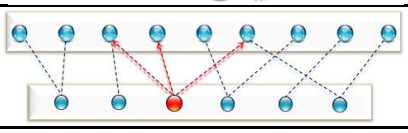
This model can give an overview about the relationship between the Student processes with School and Teacher services.

5.6 General Concept Classification

In this research, there are some important concepts which are defined and explained from the beginning of this research to now. To have a general view of those concepts, we summarized them into a table as shown below:

Table 4: Table of the general concept classification.

General concept classification	Tool used	Example	Technology applied
1) Program: • M • Mα • 3D • 5D • SD	BPM		BPMN
2) Course: • Courses of Subjects • Courses of General Education • Highly Embedded Systems Courses	BPM		BPMN
3) Area: • Theoretical Information Science • Human Information Processing • Artificial Intelligence • Computer Systems & Networks • Software Science	Component		Component
4) Subject: • Introductory • Basic • Intermediate • Advanced	Component		Component

5) Constraint: • Requirements for the completion • Some rules	Aspect (Constraint Aspect)		AO
6) Relationship between subjects: • Prerequisite	Aspect (Relationship Aspect)		AO
7) Knowledge Unit: • Some theories • Some techniques	Aspect (Extension Aspect)		AO

Chapter 6 - Conclusion and Future work

In conclusion, the research is aimed at defining a model for the educational system. We analyzed the educational system and built a general model using Service-Oriented Modeling and Business Process Modeling. We also identified and defined Aspects that are crosscutting concerns in the system by using Aspect-Oriented Technology.

The new model includes actors, services provided and used by actors, workflows which represent the way to access the services, the relationships between services, the constraints that are applied to the services and the extension services.

The building of the general model for School educational system will help the schools and teachers to build a good educational system. This will enable the students to know clearly about the services provided by the school, help them to choose suitable services and understand well about the processes they will perform during the study line based on their study plans. Therefore, the students can have a good study time in the school and be able to complete the study program successfully.

The future work includes the building of Persona Model which supports students to access services.

Bibliography

[1] IBM Web Services Architecture team. http://www.ibm.com/developerworks/webservices.

[2] CBDI Service Oriented Architecture Practice Portal.

[3] Wikipedia.

[4] S.Kemsley, "Business Process Modeling", 2007.

[5] AOSD Community. Aspect-Oriented Software Development Community and Conference. http://www.aosd.net.

[6] David J.Pearce, James Noble, “Relationship Aspects”, AOSD ’06.

[7] Ivar Jacobson, “Use cases and Aspects – Working Seamlessly Together”, Journal of Object Technology, 2003.

[8] Thomas Erl, "Service-Oriented Architecture: Concepts, Technology, and Design", Prentice Hall, 2005