

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

Nguyen Thi Hong Nhung, Koichiro Ochimizu

School of Information Science

Japan Advanced Institute of Science and Technology

ABSTRACT

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.

In this paper, 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.

Keywords

Modeling, educational system, service-oriented modeling, business process modeling, aspect-oriented technology.

1. INTRODUCTION AND BACKGROUND

All schools provide 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.

Service-Oriented Architecture (SOA) is an application architecture in which all functions are defined as independent services with well-defined invocable interfaces which can be called in defined sequences to form business processes [1]. **Service-Oriented Modeling (SOM)** is used to designing and specifying service-oriented systems with SOA.

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 [2].

But 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** [3]. By analyzing the system, we will find and define Aspects that are applied to this research.

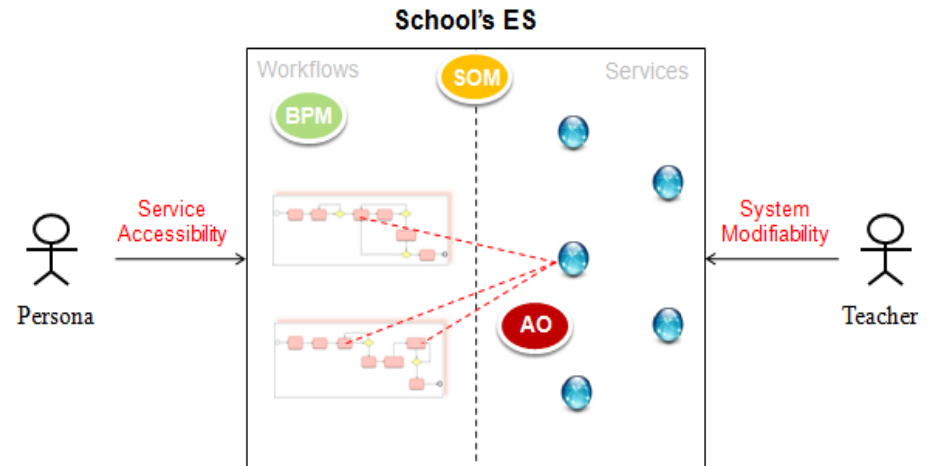


Figure 1: Research's purpose.

The primary goal of this research includes:

- Using SOM to model the system services.
- Describe several accesses to services with BPMN.
- 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.

2. BASIC MODEL OF THE EDUCATIONAL SYSTEM

Each school has a model of the educational system which includes many services, actors, classes, etc. In Figure 2, we show an example of the basic model of the educational system:

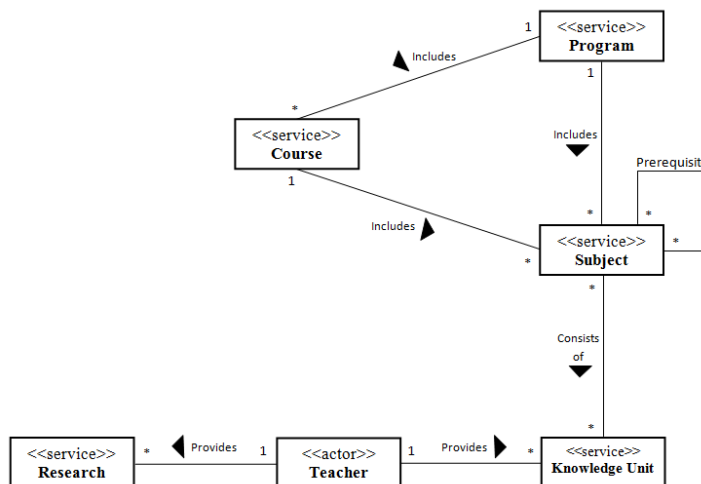


Figure 2: Basic structure of JES.

According to Figure 2, we need to understand some services 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.
- **Degree Program service:** is a top-level educational service which includes all other educational services.
- **Course service:** is a special case of degree program which includes some special subjects.
- **Subject service:** is a set of related knowledge units provided by Teachers to Students.
- **Knowledge unit:** is a Teacher's knowledge.

3. SERVICE-ORIENTED MODELING

3.1 General model of ES

The Figure 2 mentioned above represents an example of the basic model of the 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 3, we add several more classes to the basic model:

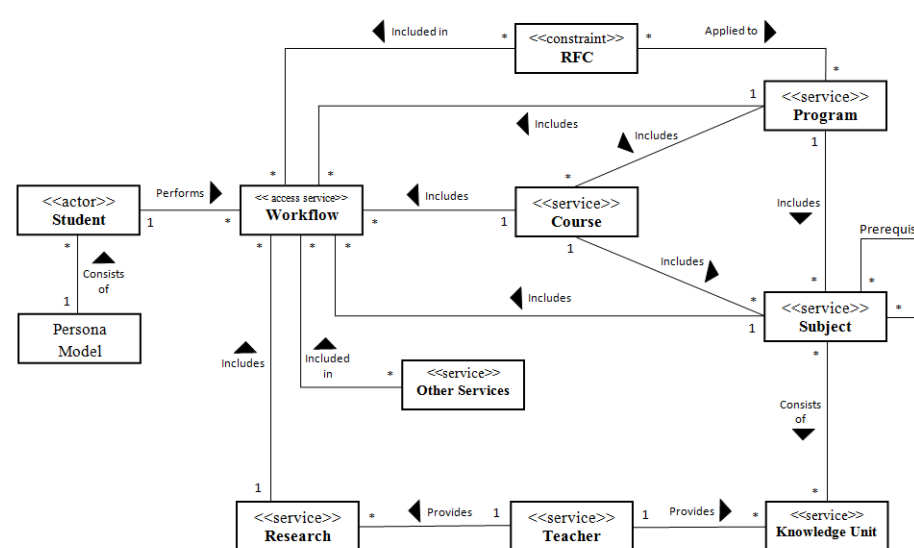


Figure 3: General structure of JES.

In Figure 3,

- **Workflow:** defines the way to access to each service which will be described in Section 3.3.
- **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.

3.2 Interaction between Student and School

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.

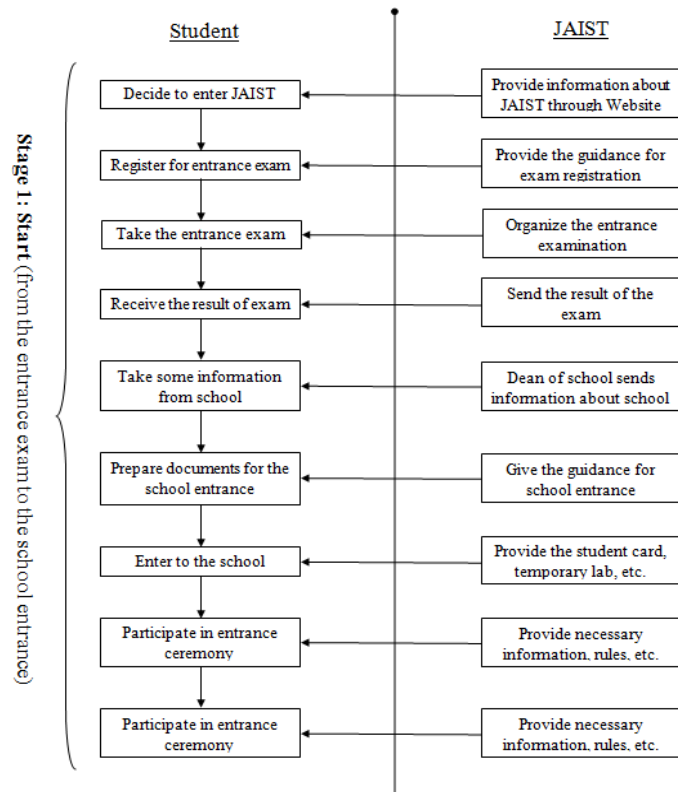


Figure 4: Example of Scenario of M Program in JES.

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.

3.3 Business Process Modeling of accessing services

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. We will show an example of created BPMs in JES.

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

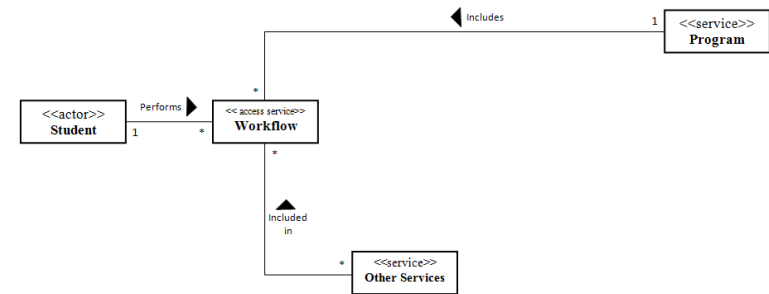


Figure 5: Program service part in the general model of JES.

And here is BPM for accessing the Program service in JES:

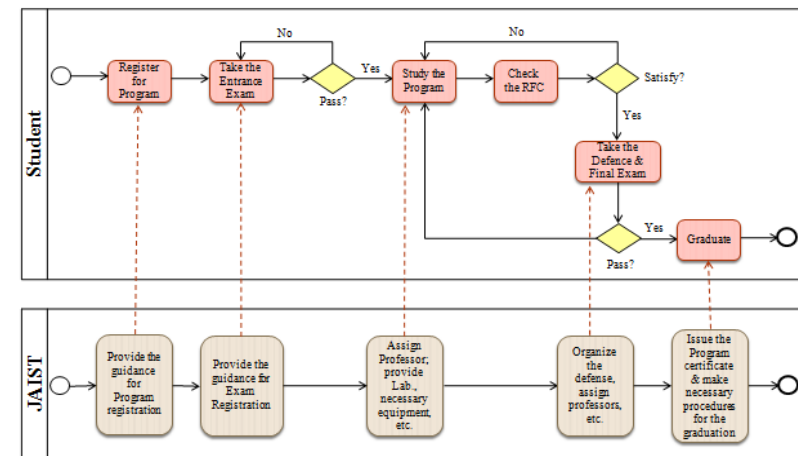


Figure 6: Example of BPM for M Program service in JES.

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 to access the M Program service.

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.

4. ASPECTS

4.1 General model of ES 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 and the constraints that are applied to School's services and processes. To build a complete model, we also include those Aspects in the model as illustrated below:

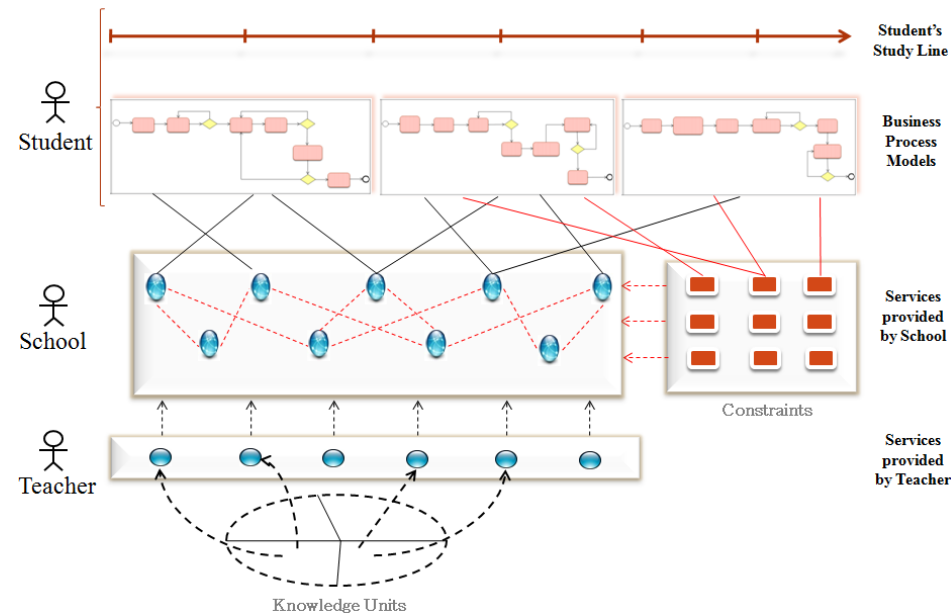


Figure 7: The general model of ES (with Aspects).

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

4.2 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. They enable teachers to revise the curriculum easily. After making some analyses, we find out that in our research, there are two types of Aspects:

- Constraint Aspects
- Relationship Aspects

4.3 Constraint Aspects

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

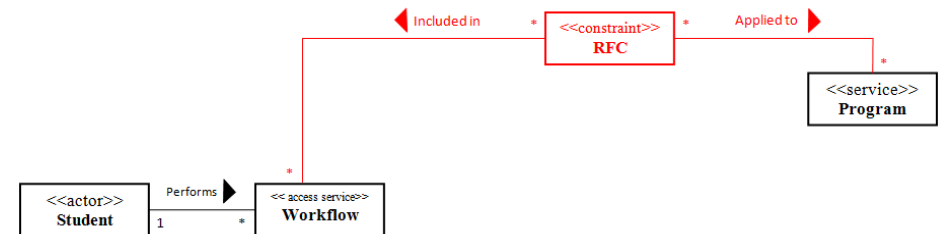


Figure 8: Constraint part in the general model of JES.

4.3.1 Definition of Constraints

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

4.3.2 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 subject 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.

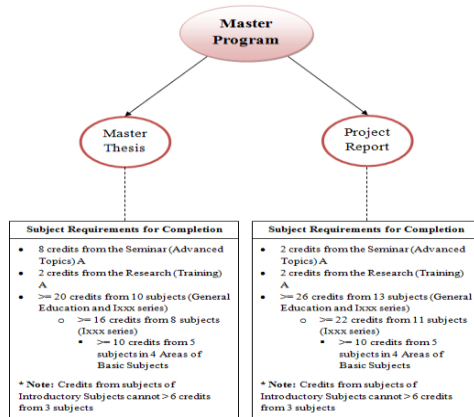


Figure 9: Example of Constraints in JES.

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

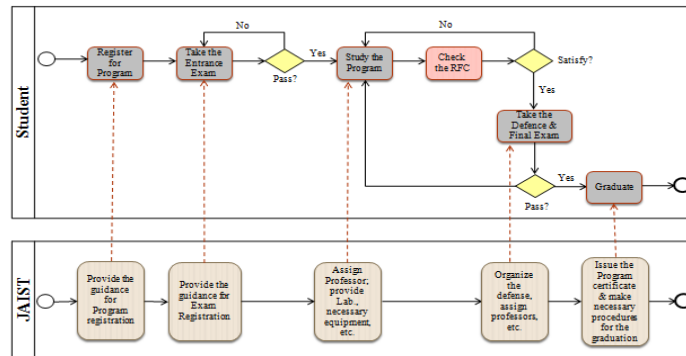


Figure 10: Example of Constraints in BPM.

4.3.3 Constraint Aspects

Moreover, the same RFC can be applied for different services (as shown in the Figure 11) and they may crosscut those services.

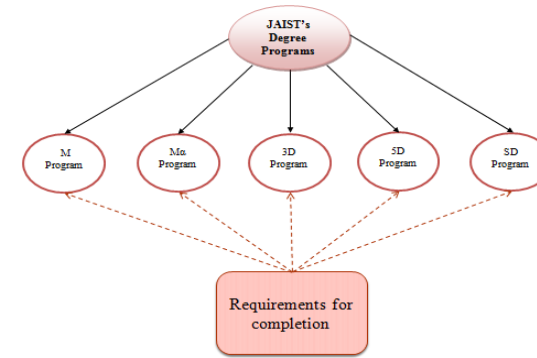


Figure 11: Example of RFC in JES.

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

4.3.4 Structure of Constraint Aspects

The figure below shows the structure of a Constraint Aspect:

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

Figure 12: Structure of CA.

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

<<aspect>> CheckRFC
<<pointcut>> M.StudyProgram Ma.StudyProgram 5D.StudyProgram
<<advice>> checkRFC()

Figure 13: 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 process them independently so that they can be easily updated, changed or removed.

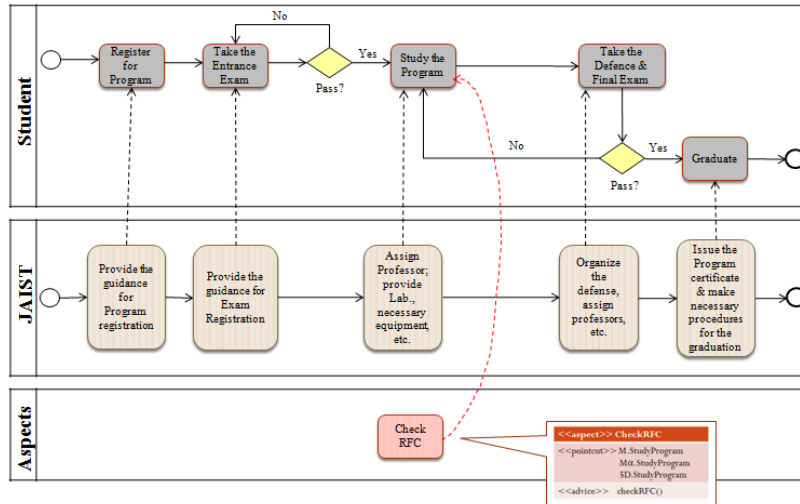


Figure 14: Example of Constraint Aspects in BPM.

4.4 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 [4]. 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:

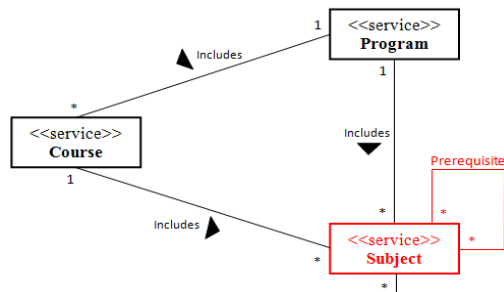


Figure 15: Relationship part in the general model of JES.

4.4.1 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 I212 subject, you must take the I114 subject first or if you already took the I616 subject, then you cannot register for the I212 subject. The figure below represents the relationships between subjects which are called Prerequisites. There are two types of Prerequisites: “Needs” and “Cannot register”.

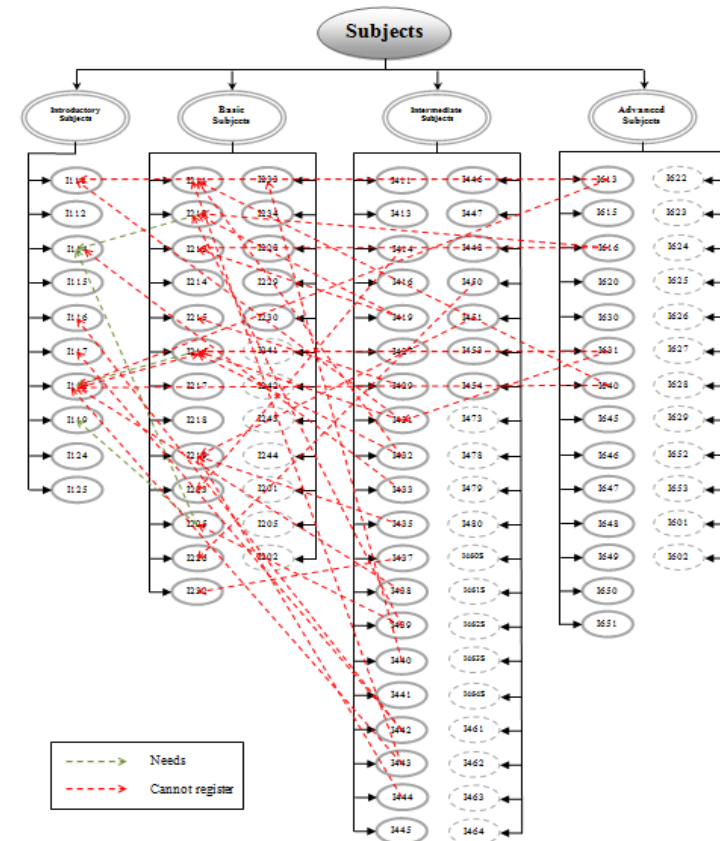


Figure 16: Relationship between Subjects in JES.

4.4.2 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.

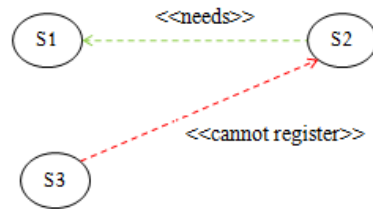


Figure 17: Relationships between S1, S2 and S3.

To deal with these relationships, usually we will store their information in S1, S2 and S3 as shown in the Figure 18:

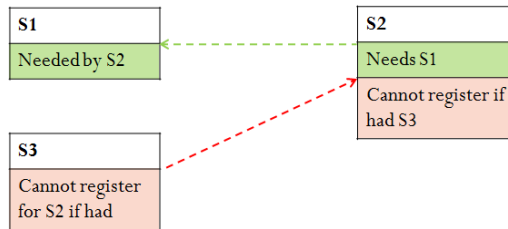


Figure 18: Information about relationships is stored in each subject.

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 [4].

We can see that a relationship is shared in at least two objects or in other words, it crosscuts across those objects. Therefore, we can treat them as separate concerns that crosscut 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.

4.4.3 Structure of Relationship Aspects

The figure below shows the structure of a Relationship Aspect:

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

Figure 19: 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 20:

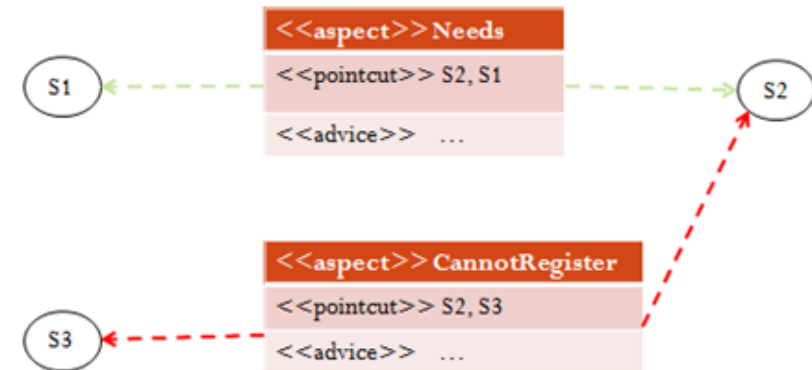


Figure 20: Example of Relationship Aspects & their structure.

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 pointcuts as follows:

Table 1: 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
	I631, I216
	I631, I431
	I640, I118
	I640, I211
Aspect Name	Pointcut
<<aspect>> CannotRegister	I118, I113
	I118, I216
	I216, I222
	I232, I436
	I233, I426
	I234, I425
	I441, I428
	I442, I224
	I444, I641

5. 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 and the constraints that are applied to the 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 the students to access to the services.

6. REFERENCES

- [1] IBM Web Services Architecture team. <http://www.ibm.com/developerworks/webservices>.
- [2] S.Kemsley, "Business Process Modeling", 2007.
- [3] AOSD Community. Aspect-Oriented Software Development Community and Conference. <http://www.aosd.net>.
- [4] David J.Pearce, James Noble, "Relationship Aspects", AOSD '06.
- [5] Thomas Erl, "Service-Oriented Architecture: Concepts, Technology, and Design", Prentice Hall, 2005.