



## Service Oriented Analysis and Design with Educational Information System Management

Prof. (Dr). R. Kamatchi

Assistant Professor, KJSIMSR,  
Mumbai, India

**Abstract:** *The Service Oriented Analysis and Design is a new paradigm into the Software development model. This is a meet-in the - middle architecture which enables the complete process of application building simple and flexible. This paper is comparing the SOAD model with OOAD model and highlights the advantages of SOAD model in the case of Educational Information System. The first section of this paper describes the basic architecture of EIS model. The second section deals with the basics of SOA and SOAD. The third section compares the basic OOD model with the SOAD. The fourth section explains about the detailed SOAD design for the examination module. The final section describes the process choreography with the benefits of SOAD model.*

**Keywords:** *Service Oriented Analysis and Design, Object Oriented Analysis and Design.*

### I. INTRODUCTION

The increased utility of the services coupled with the increase in functionalities. This enables the use of Service Oriented Architecture in various sectors. Traditional software development models are implementing either top down or bottom up approach. The SOA starts with the service identification, the middle of the entire architecture and enables the need for a Service Oriented Analysis and Design (SOAD) supports the meet-in the -middle architecture, and provides the consumers with extensive customization facilities.

### II. METHODOLOGY

The main focus of this paper is developing a case of educational system through which the detailed study of the meet-in-middle architecture can be realized. As an initial step of this analysis, the samples of 100 common services among the educational institutions were identified with its own process and the operation description. After defining the model, this paper is discussing about the implementation of SOAD model with the Educational Information System.

### III. BACKGROUND

#### 3.1. Educational Information System:

The EIS model considered for the analysis contains

- 1) Web client
  - a) Educational Service Center
  - b) Educational Service Providers
- 2) Educational Server
  - c) UDDI

The web clients are the service consumers. The service consumers may vary from the students, teachers, parents to placement agencies. For example the students query to access the books from the e-library will be forwarded to the educational server which will then be directed the service registry. The part of the keyword will be considered for the search and the *e-bookaccess* service will identify the content and distribute the same to the user through the interface. The interfaces can also be called as a web brokers who will enable the core character of SOA- Interoperability. The interfaces will help the different consumers to communicate from their own setup to a different application setup.

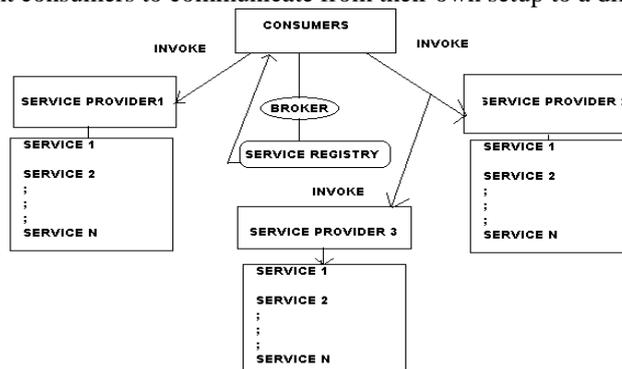


Figure1: EIS model

The educational server utilizes and manages the distributed education resources. This contains three components.

1. Server registration
2. User registration & profile updating
3. Content discovery

### **3.1.1. Service registration:**

When a new service is created by the service providers then it should be registered with the service registry and the invoking descriptions will be updated by using WSDL (Web Service Description Language). This will make the service to be available from anywhere to multiple service consumers. This can be invoked based on the consumer's request.

### **3.1.2. User registration and profile updating:**

This enables the users to have access with the services available through the network. Before invoking any service from the registry, the user should register his profile and update the same during the subsequent access.

### **3.1.3. Content Discovery:**

This carries the consumers query and searches the same from the registry. The service will be identified from the registry and can be invoked from the corresponding service provider.

The above mentioned architecture can be connected with the public server using the internet application protocols like HTML, XML, and SOAP etc.

## **IV. COMPARISON BETWEEN SOAD AND OOAD MODEL**

The OOAD model normally rotates around the objects. Everything is defined in terms of classes and objects which are bundled with its own properties and methods, in turn OO model deals with a low level of abstraction. The OOAD model generally supports the complete life cycle like analysis, design and development wherein the development step with OO model goes around the classes.

Important properties of OOAD:

1. Encapsulation
2. Information hiding
3. Classes & instances
4. Association & Inheritance
5. Information Abstraction
6. Polymorphism

The classes are the lowest level of abstraction in the overall development cycle. The inheritance property makes the system more tightly coupled and provides an increased dependency among the objects. OO is a valuable approach for design of the underlying class and component structure within a defined service.

In contrast with the OO model, the SOAD follows the meet in middle architecture. Every service is completely encapsulated and complete in nature. Every service design philosophy is the same which is self describing in nature because of its simple naming theory. Every service is stateless in the prescribed problem domain. The services can be chosen dynamically and choreographed to create a complete process. The standard legacy application can be compensated by the service repositories and the service choreography process.

For the simple and agile service technology, the services can be categorized and aggregated according to their usage. The service grouping can be made based on their usage and purpose. Generally the services are divided into software and business services group. This grouping makes the process of service identification easy and simple. Every service contains the syntax, semantics, QOS characters and interface contract (SLA).

## **V. SOAD MODEL**

The main ingredients of SOAD are OOAD, BPM and EA. This only cover part of the requirements needed to support the SOA paradigm. While the SOA approach reinforces well-established, general software architecture principles such as information hiding, modularization, and separation of concerns. It also adds additional themes such as service choreography, service repositories, and the service bus middleware pattern, which require explicit attention during modeling. [1][2]

The SOAD model is not completely replacing the existing OO architecture. It considers the important aspects of Object Oriented Design and within the modules of the OOD; the SOA is finding its way to get placed. The important concepts of OOAD like Encapsulation, Information hiding, Associations, inheritance etc are also utilized with the Service Oriented Design model with the simple replacement with its meaning and the placement.

The bottom level of OOD model contains the objects, methods, classes etc. The same is replaced with the services, processes and the activity list in SOAD structure. A process in SOAD can be compared with the class in the OOD architecture. A process comprises of various services and the policies bundled together. The services are the core components of the processes which can be used to perform certain activity wherein the policies and the semantics are the methods which help the consumer to invoke the services. [3]

OOD model	SOAD model
Functional domain	Process(Examination)
Business process	Conducting examination(activity)
Business services	Service description(WSDL)
Software services	Individual services(Result, online exams)
Software components	Operations(Validation, Grading)

**5.1. Characteristics:**

- The SOAD model is a simple architecture which contains the components in every level in a transparent way.
- The structure can be prepared on runtime by choosing the services independently and combined to produce the complete system architecture.
- Any addition and deletion of services within the process choreography can be performed independently without affecting the existing working environment.
- Well-crafted services bring in flexibility and agility to the business; they facilitate ease of reconfiguration and reuse through loose coupling, encapsulation, and information hiding.
- Well-designed services are meaningful and applicable for more than enterprise application; dependencies between services are minimized and explicitly stated.
- Service abstractions are cohesive, complete, and consistent
- A frequently stated assumption is that services are stateless (for example, not conversational); this statement would be weakened to request services to be as stateless as possible in the given problem domain and context.
- The service naming is understandable for domain experts without deep technical expertise.
- In a SOA, all services follow the same design philosophy (which is articulated through patterns and templates) and interaction patterns; the underlying architectural style can easily be identified (for example, during architecture reviews).
- The development of the services and service consumers requires only basic programming language skills in addition to domain knowledge; middleware expertise is only required for a few specialists, that in an ideal world, work for tool and runtime vendors, and not for the companies crafting enterprise applications as SOAs.

**5.2. Components:**

The definition of SOAD model contains the following four phases wherein with each phase, the development of the design gets tuned up and the final formalized structure can be derived.

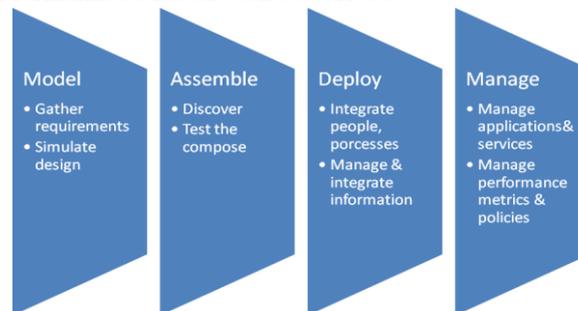


Figure 3. SOAD Components

**5.2.1. Model:** This phase collects the requirement specifications from the consumer and models the suitable services with each requirement specified. After the accumulation of requirements and modeling, the simulation of the complete structure can be prepared in order to get the basic idea of the architecture.

**5.2.2. Assemble:** Once the requirement model is prepared, then corresponding services which performs the required task are identified and testing the same can be performed. A locator can perform the task of service identification and the verification of the same. This phase is preparing the full proof model of services with verification and testing.

**5.2.3. Deploy:** The deployment phase describes the integration of various security parameters with the required specifications. The integration of services in a specific process and the application environment is possible only when they are compatible with each other and the security parameters should be embedded with each service. The federated interface can be used to provide the connectivity among the similar user groups.

**5.2.4. Manage:** This phase is enabling the architecture to manage the processes and the individual services in the application environment. The management of the performance level is an important feature of the proposed architecture in order to provide sustainability to the process execution. This helps the workflow to perform the activity successfully without any further verifications or the analysis. Any degradation in the performance level is supplemented by another service with the same specifications. This maintenance is possible only by the workflow database present in the EIS security model.

**5.3. Service classifications:**

With the SOAD framework, the services present in the proposed model can be classified into the following categories.

1. Educational services: Supports the educational purposes. The services like form filling, mark sheet generation can be classified into this category.
2. Management services: These are the services which helps the system to maintain the service collaboration within the application environment which also takes care of the security maintenance. Example: SAS, AAA services.
3. Process services: These services help the system to have a correlation and collaboration among the services. This enables the smooth process choreography.
- Example: Orchestration services, service brokers
4. Interaction services: These services are enabling the interaction among various services and processes in a single working environment.
- Example: Interfaces & federated services.

**VI. SOAD MAPPING**

Once the services are classified into various groups then the identification and modeling task can be made simple. Once the services are modeled then it can be correlated to the existing OOD model. The following diagram is indicating the way the services can be modeled to various steps involved in the traditional OOD model. In the following diagram a single service named Examination is considered for the analysis. In the same way various other services involved in the EIS model can also be defined.

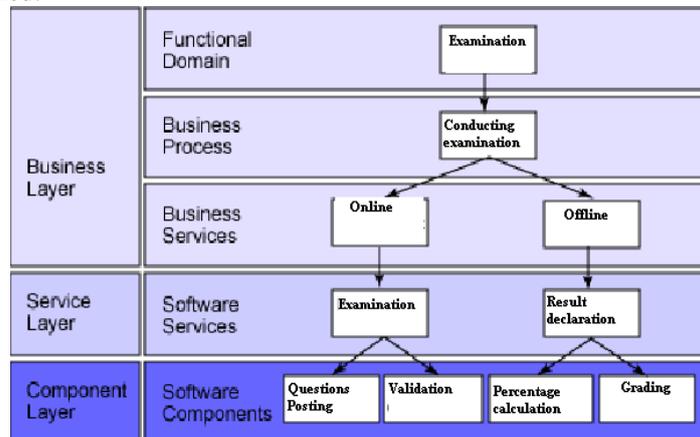


Figure 4. SOAD Mapping

**6.1. Sample SOA design components:**

The following diagram is explaining about a sample service oriented model where only one process is considered and various components of the same are expressed in detail. In the Object Oriented Design, each and every object is complete in nature which contains its attributes and the methods bundled with the same. The same way in Service Oriented Model, the services are complete in nature which contains various components bundled together and provides the required component details to different set of users.

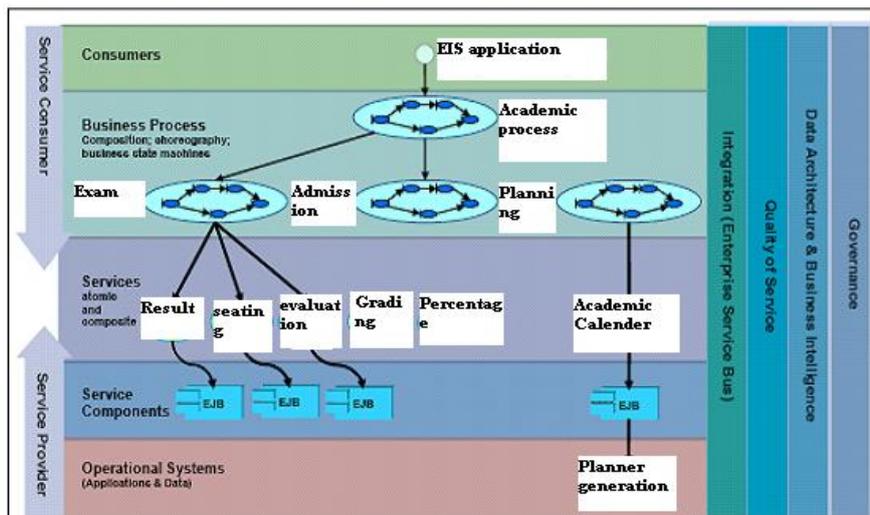


Figure 5: Sample SOA design components

**6.2. Components:**

Consumers: The major part of the consumers with EIS model deal with the educational application which is kept in the higher level of the application architecture.

Process: There are various processes defined with the proposed model which are mainly concentrating on the academic domain.

In the academic domain there are a list of processes defined wherein the three sample processes are defined in the above figure for representation.

Services: The services may be atomic or composite. The atomic services are used for the complete unique purpose. The composite services work in combination with the related other services.

Atomic services: Seating arrangement

Composite services: Grading, Percentage calculation

Operational system: This is the final output which combines the data with the services identified and produces the desired output. [4][5]

## VII. SOAD MODEL

The following case describes the way to develop a SOAD model in connection with the OOAD and BPM model. The Examination process is considered for the same.

The Mark sheet preparation is a process domain under the Educational Information System by which the student's evaluation is performed. The examination process represents the evaluation activities such as online quiz, assignment evaluation, conducting term exams and evaluating and updating student's performance frequently.

The business scenario is as follows:

- The mark sheet is prepared whenever the student enroll for an examination.
- For each examination activity or operation a separate mark sheet is generated with the personnel and the performance details.
- The complete student record would be verified before the examination schedule like fee default, attendance record, examination registration records etc.
- A seating arrangement and room allotment should be scheduled before scheduling the examination.
- Individual registration is performed.
- The enrollment and the eligibility testing are performed and the student performs the exam and the evaluated mark sheet is prepared, otherwise the mark sheet generation doesn't take place.
- When the student enrolls, the sequential activities take place.
- The actual performance details are registered and recorded.
- All the other grades related to the final calculation like internal assessment, sports incentives, NSS appreciation needs to be recorded.
- On completion of all evaluations, the consolidated evaluation report is generated.
- A mark sheet is generated and presented to the student.

The following diagram explains the macro flow of the mark sheet generation process.

The mark sheet generation process can be performed using the OOD model wherein the development of interface among the various related objects needs external policy implementations. As the system is tightly coupled, the changes in the rules and the order of processing may bring difficulty in the development of the complete model. [6][7][8]

For example a student belongs to science stream need to submit the assignment regularly to fulfill the evaluation criteria completely. But the student with the commerce stream can completely skip the assignment submission which is not considered as a part of the whole evaluation process. This kind of variations in the rules (semantics) needs to be externalized. The following diagram explains the OOD model for the case considered.

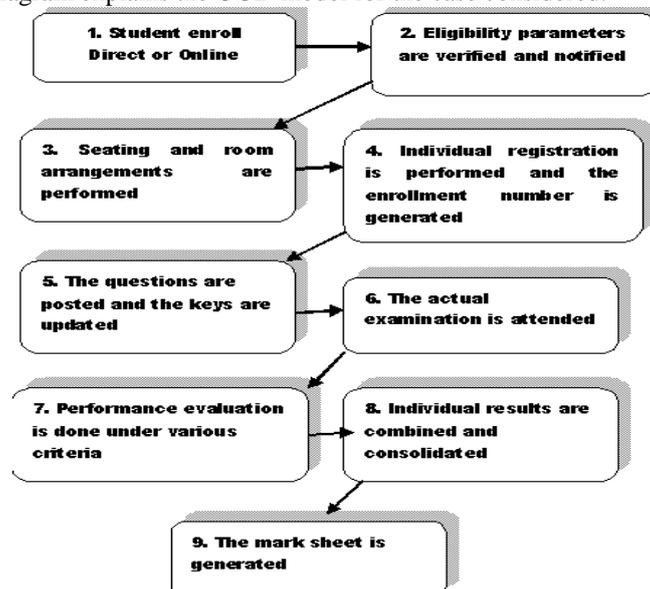


Figure 6. Macro flow

The SOAD provides an excellent solution to the issues addressed by the previous model. Since it groups the services on the basis of related behavior, rather than encapsulated, the set of services will be subtly different from a business object model. [9][10]

The SOAD model does not represent a functional system. There is no sense of flow, nor description of business events or rules. In the SOA paradigm, business process choreography is maintained externally and responsible for the sequence and timing of the execution of services. The following diagram explains the BPM for the mark sheet generation.

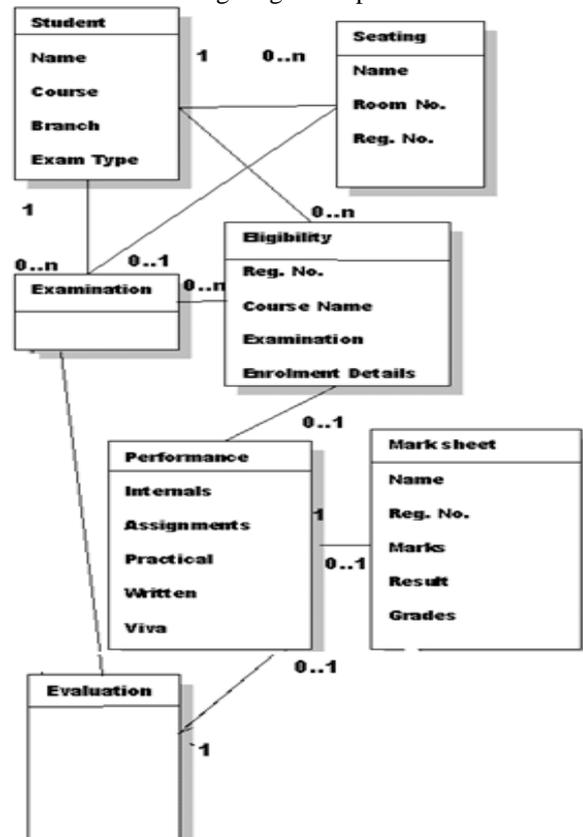


Figure 7: OOD Model

The above diagram clearly depicts the relativity among the classes. Every class comprises its own attributes. The changes made with any single class may interfere with the whole hierarchy. This clearly indicates that the whole system is tightly coupled. As this system is rigid within its design, any addition or deletion with the existing system would be difficult. In contrast with the SOA model, this OO model doesn't support the agile development methods. The development model is static and complicated because of its inheritance. [11]

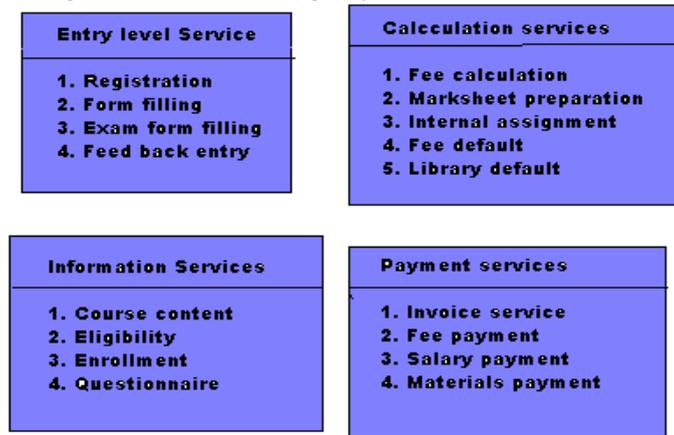


Figure 8: BPM model

The above figure clearly illustrates the business process which was prescribed in the beginning of this paper. Every activity in the pipeline has an effect to the next. Based on the results of the previous activity the process would continue and the corresponding services for the same would be adopted.

**VIII. SERVICE ORIENTED MODELING**

The similar structure with the service oriented modeling can be developed in the following way. The services related to the mentioned scenario can be categorized in the following way.



The considered examination process workflow can be indicated as follows

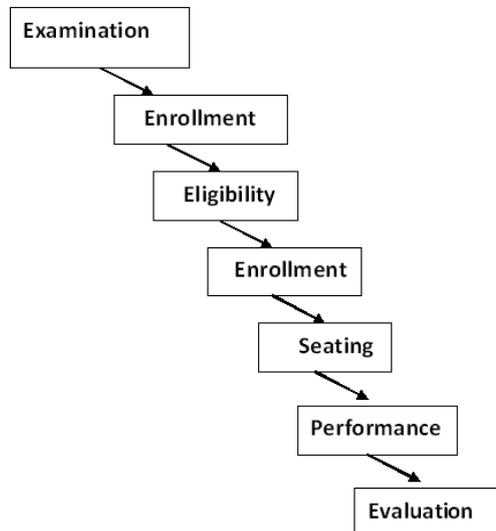


Figure 9: Examination workflow

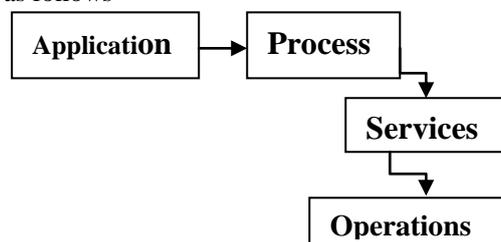
The examination process contains 6 activities which need to be performed in a sequential way to get the desired result. Every activity can be replaced by a service. Every service is complete in nature and designed by different service creators. The services participating within a process may be belonging to different service providers. The identification of services is performed on the basis of the requirements collected from the user and the operations mentioned by the service creators.

Operation listing:

Enrollment: 1. Course name 2. Roll number 3. Exam code 4. Paper name 5. Date	Eligibility: 1. Personnel details 2. Percentage 3. General details 4. Other details
---	---

The requirements specified by the user would be compared with the operations and the specifications of the services and the appropriate services would be identified.

The SOAD design can be described as follows



The system development with SOAD begins with the service identification which is in the middle of the hierarchy. Once the services fulfilling the activities are identified and their operations are compared with the specifications. Then the appropriate single service would be defined. The collection of these services would describe the individual process. The collection of all these processes leads to the EIS application definition.

he process choreography is the next level Service Oriented Analysis and Design model wherein the relativity and the reusability among the services can be analyzed and tabulated. The choreography explains the meet in middle architecture of the SOAD. Every process design starts with the identification of services based on the requirements specification collected from the users. Every requirement is mapped with an operation of the service.

### IX. PROCESS CHOREOGRAPHY

The following diagram gives the sample of process choreography of a simple examination process. This diagram gives the pictorial representation of only 5 services and their corresponding operations. The sample application contains 10 processes and every process contains various sequences of activities.

Even though the services participate in a single process, they are independent and can be removed and attached based on the technical requirements. This is making the SOAD model loosely coupled and flexible.

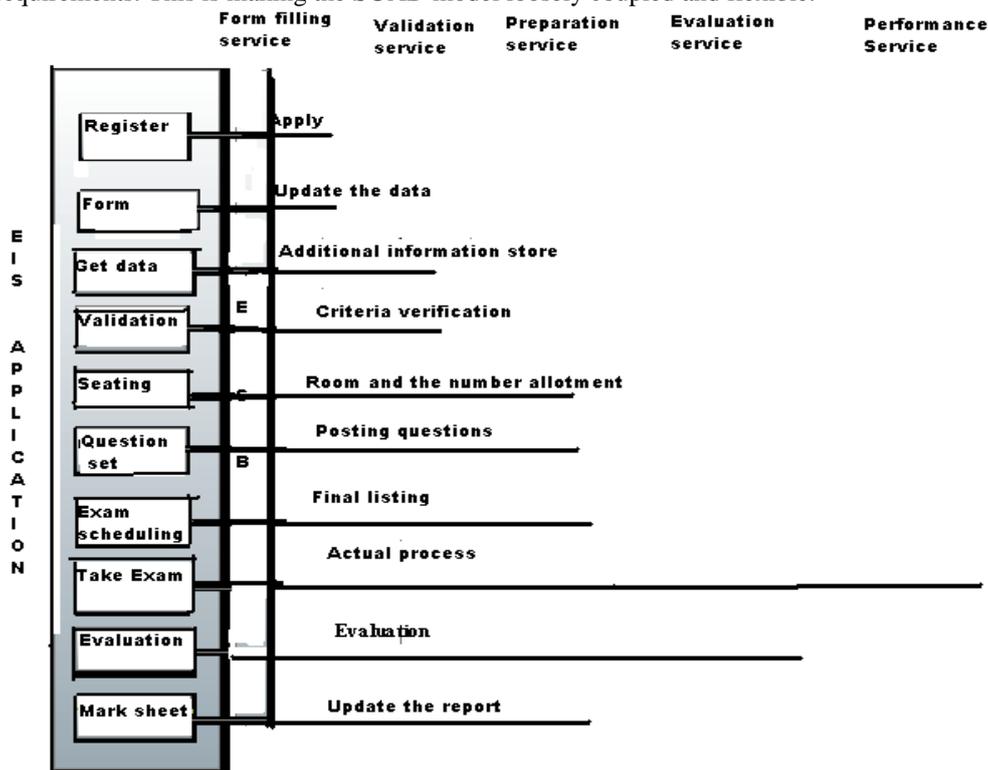


Figure 9: Process choreography

The above mentioned diagram clearly depicts the designing and development principles of SOA. Every service used with this modeling is unique and complete in nature. These services are clearly reused by many consumers in different set of requirement specifications. The similar development with the OO model has to begin with the classes and their objects along with the relativity among the classes. This may restrict the free flow of functionalities among the services.

### X. ADVANTAGES

The services that are participating in a single process are dynamically collected. The process developed with SOA model lasts only for few seconds. The various states of the process is stored in the data base for the further activation. The process choreography focuses on the transition between states. The service reuse by more than one process is possible only by SOAD. For example the choreography stated in this paper clearly explains about the reusability of the services among various process descriptions. SOAD provides loose coupling and less coherence because the services participating in a process is identified and grouped dynamically. The individual services are complete in nature wherein, it works independently in the whole process choreography.

### XI. CONCLUSION

This study is providing a detailed insight of SOAD model and its possible implementation with the Educational Information System. The comparison of SOAD model in accordance with the traditional OOAD, SSAD models proves the meet in middle architecture clearly and defines that this SOAD model can be build upon on the well-established and proven OOAD, EA and BPM components. The OOAD model can be mapped into the SOAD model easily and proves the higher reusability component of the later model. It is using the existing models and providing a new dimension for the development.

**REFERENCES**

- [1] IVAR JØRSTAD, SCHAHRAM DUSTDAR, “A Service Oriented Architecture Framework for Collaborative Services”, *Proceedings of the 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise (WETICE'05)*, pp..1-8.
- [2] AMELIA MAURIZIO, JAMES SAGER, “Service Oriented Architecture: Challenges for Business and Academia “, *Proceedings of the 41<sup>st</sup> Hawaii International Conference on System Sciences – 2008*, pp... 1-7.
- [3] ERIC PULIER, HUGH TAYLOR, “Understanding Enterprise SOA”, *Manning Publications CO.*, 2006. Pp... 23-25, 139-150.
- [4] L.CHERBAKOV, R.HARISHANKAR, S.KALYANA, “Impact of service orientation at the business level”, *IBM Journal*, Volume 44, Issue 4, pp... 2.
- [5] PASCAL BOU NASSAR, YOUAKIM BADR, “Towards Integrating Security Services in e-learning Platforms”, *IEEE 978-1-4244-3834-1/09*, pp... 573-576.
- [6] K.GOTTSCHALK, S.GRAHAM AND J.SNELL, “Introduction to Web service architecture”, *IBM Journal*, Volume 41, Number 2, 2002, pp... 3, 4.
- [7] N.CH.S.N.IYENGAR, JAGHADEESH.R, KARTHIK.T.U, PRAKASH.N.M, IJAYASHERLY.V, “Distributed and Learner Adaptive E-Learning System Using Service Oriented Architecture”, *IJCSNS International Journal of Computer Science and Network Security*, VOL.9 No.6, June 2009, pp..187-190.
- [8] GANESH, D. M. A. J., "Web services and flexible business processes: towards the adaptive enterprise," *Information & Management journal*, vol. 42, pp. 921–933, 2005.
- [9] THOMAS ERL, “Service-Oriented Architecture “, *Prentice Hall PTR*, 2004, pp.. 35.
- [10] OLAF ZIMMERMANN, “Elements of Service-Oriented Analysis and Design”, *IBM journal*, 02 Jun 2004.
- [11] RAMARAO KANNEGANTI, PRASAD CHODAVARAPU, “SOA Security”, *Manning Publications Co*, ISBN: 1-932394-68-0, Pg.No-08-09.