An Automated Service Realization Method

Mahshid Marbouti[1], **Fereidoon Shams**[2]

[1] **Electrical and Computer Engineering Faculty**
Shahid Beheshti University GC, Tehran, Iran

[2] **Electrical and Computer Engineering Faculty**
Shahid Beheshti University GC, Tehran, Iran

Abstract

One of the key activities in service oriented analysis and design is service modeling. The last and most important step of service modeling is service realization, which should be done after service identification and specification steps. Since the output of realization step is the input of implementation step and organization strategic decisions on services are being made in this step, realization is a very important step. Existing methods for service realization ignore the automation capability. They provide human based prescriptive guidelines which mostly are not applicable at enterprise scale. This paper proposes a novel method called Automated Service Realization Method (ASRM) that automatically realizes service model. This automation focuses on activities needs human intelligence such as component identification and class diagram extraction;

Keywords: *Service Oriented Architecture, Service Modeling, Service Realization.*

1. Introduction

Service Oriented Architecture (SOA) is a new approach in software architecture. It has created a bridge between business and information technology. In this style of architecture a business model will be mapped to software services [1]. These services interact to meet the business requirements. In all SOA methodologies and life cycles, the first phase is service modeling or service analysis.

Service modeling includes three steps: service identification, service specification, and service realization [2]. During the service identification step, the business model will be mapped on software services. During the service specification step, operations will be assigned to these software services [3]. The Output of this step is service model, which consists of services, their operations, and messages that are delivered among them. In the final step realization decisions will be made for software services and the service model will be converted to the component model [4].

The IBM methodology Service Oriented Solution Lifecycle (SOMA) [2], discusses the main activities about service oriented modeling and architecture. It addressees service identification, specification and realization techniques. Although many research studies have been conducted on service computing, there are limited ones on service modeling and they do not have the automation capability. The ASER group has been worked on proposing a framework for automation of service identification, specification and realization activities. The Name of the framework is Automated Service-Oriented Modeling Framework (ASOMF), the input of the framework is the business model and the output is the realized service model [5].

There are three main automated methods in ASOMF:

Automated Service Identification Method (ASIM): Identifies software services from business model. The method gets the business model and the output artifact of the method is the analysis level service model.

Automated Service Specification Method (ASSM): specifies software services from analysis level service model. By giving the service model, ASSM, transforms it to the design level service model.

Automated Service Realization Method (ASRM): is the main subject of this paper. We can define it as a method for making architectural decisions. ASRM transfers the service model to the component model by identifying the components and extracting class diagram for each component. ASIM is about high level design but ASSM and ASRM is about detail design.

The rest of this paper has been organized as follows: section 2 introduces the most related work, section 3 describes the ASRM method, and section 4 evaluates the methods based on expert outputs. Finally the conclusion is described in section 5.

2. Related Work

We can divide service realization methods into two main categories. The first category consists of methods, which are based on SOMA methodology [2]. In these methods, service realization is a phase in which realization decisions and mapping to components should be done. Methods, which are proposed by Johnson [6], Portier [1], Amsden [4], Arsanjani et al. [2], Zhang et al. [7], are from this category. Johnson's [6] method does not cover the full realization of service model and its elements; moreover, it does not mention how to get them done automatically. Portier [1] uses model driven approaches for designing service model but does not propose a procedure for constructing such a model. Amsden [4] focuses on the description of service model's elements, which should be realized in service realization phase, but does not consider the process of designing service model and realizing the components. Arsanjani et al. [2] explain service realization phase thoroughly, but do not describe how to achieve the artifacts produced in this phase. Therefore, by ignoring the technical metrics, the automation of the realization process would not be achieved. Zhang et al. [7] have introduced a modeling environment namely SOMA-ME which acts as a framework for model-driven design of SOA solutions using SOMA method.

Second category consists of the methods, which are based on specific service-oriented solution lifecycle such as Erl [8], and Papazoglou [9]. They have introduced a thorough methodology for service-oriented solution development based on prescriptive guidance. Although there are differences between our point of view and theirs about service modeling, the adoption of technical metrics and also automation capabilities have been ignored in their work. Decastro[10] et al. also proposed another method which model driven but not automated. There are also class diagram extraction methods based on NLP which are not using service oriented methodology [13-14].

Automation, adopting technical metrics, and utilizing model-based principles are the most important capabilities of the realization method, which is proposed in the present paper.

3. Main Part

3.1. Inputs

The CRUD Matrix as an input of the method is described below.

The CRUD matrix mediates between enterprise business model and enterprise service model in order to extract candidate services populates the enterprise service model [5]. The mediated model is formed as matrix, which has EBPs as rows, BEs as columns, and semantic relationships (Create-C, Update-U, Delete-D, and Read-R, with the priorities as C>U>D>R) as cells:

"C" means the EBP CREATES the BE.
"R" means the EBP READS the BE.
"U" means the EBP UPDATES the BE.
"D" means the EBP DELETES the BE.

In order to compute the value of each technical metrics, the model should be transformed into value-based rather than being tag-based. Therefore, the corresponding value (weight) of each tag should be replaced in the model according to 1=>C>U>D>R=>0 [11]. The larger the weight in the entry, the closer the relation between EBP and BE. In this paper, we assume that each value in the CRUD matrix is a real number between 0 and 1. In this paper, due to simplifying computations, we adopt these substitutions: C: =1, U: =0.75, D: =0.5,R: =0.25.

Figure 1 represents a sample CRUD Matrix.

	BE1	BE2	BE3	BE4	BE5	BE6
EBP1	C	U			U	
EBP2	R	R				
EBP3			C	U		R
EBP4			R	C		
EBP5			U			
EBP6			U	R		
EBP7					C	
EBP8					U	R
EBP9		R				U

Figure 1. A Sample CRUD Matrix

3.2. Step1: Component Identification

The goal of the first step is to transform the service model to a component model. So the aim of this part is to propose a method for realizing the services components. The more intensity of semantic

relationships between two services, the more the probability of realizing them to one component [12]. Therefore identification of components can be done by measuring the intensity of semantic relationships whereas this intensity can be logically deduced from CRUD matrix as the input of introduced approach.
To simplify the calculations, each relation in the CRUD matrix has given a weight which shows the intensity of the relation. This weight is based on the relation type, which can be "Read" (R), "Update" (U) or "Delete" (D). General assumption is that if a cluster contains "C", the cluster always represents a single service. The correlation intensity degree can be used as a quantitative measure of service coupling and cohesion. Besides, the number of relations between services can be also used in the process of grouping services into rational distinct categories; so that each category can be considered as a realized component.
Imagine I(a,b) indicates intensity of one relation between "a" and "b", in which a and b are representative of identified services. In fact a and b are two clusters of the input matrix, if the number of relations between a and b is equal to "n", then the ultimate intensity of the relation is defined as equation (1):

$$T(a,b) = \sum_{i=1}^{n} I_i(a,b) \quad (1)$$

On the other hand, the intensity of a self-relation is equal to zero:

$$T(a,a)=0 \quad (2)$$

So if "Si" and"Sj" are two services and T(Si,Sj)>α, then Si and Sj can be realized as a single component. Parameter α is calculated as equation (3) :

$$\alpha = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} T(a_i, a_j)}{N} \quad (3)$$

In the above formula "a_i" and "a_j" are two services, "n" represents the number of services and "N" indicates the total number of all possible relations, which is calculated as equation (4):

Figure **3** illustrate the component identification algorithm and class diagram extraction algorithm.

$$N = \frac{2^n - n}{2} \quad (4)$$

3.3. Step 2: Class Diagram Extraction

The next step in ASRM, is to design each component by a class diagram. To design a class diagram the three steps are necessary: Class Identification, Class Specification and Identification of relation between classes.
Class Identification: The first step in designing a class diagram is to identify classes from business elements. The more coupling between two business elements, there is more probability of realizing them into one class. Therefore identification of classes can be done by measuring the coupling between classes; this coupling can be logically deduced from CRUD matrix as the input of introduced approach. By the amount of coupling, the business entities can be grouped into distinct categories as realized classes.
Equation (5) shows how we can measure the coupling between two business entities via CRUD Matrix:

$$Coupling(BE_i, BE_j) = \frac{\frac{\sum_{Method \in M_{BE_i \cap BE_j}} I(Method, BE_i)}{Int(P_{BE_i})} + \frac{\sum_{Method \in M_{BE_i \cap BE_j}} I(Method, BE_j)}{Int(P_{BE_j})}}{2}$$

- $Int(P_{BE_i}) = \sum_{j=1}^{n} I(Method_j, BE_i)$
- *I (Method, BE) shows the semantic relationship between a method and a business entity.*
- *Indicates set of methods which have a semantic relationship by* BE_i. (5)

Class specification: The next step in class diagram design is class specification. In this step we should assign a method to each class. The assignment should be based on intensity of relation between a method and BEs which are contained in the class.
Identification of relation between classes: If a method creates a BE, then if there is any BEs which have a relation with that method, then we can assumes those BEs have a semantic relationship with each other.
In the Figure 2 and

Specify Service Components

Input: **Clustered CRUD Matrix**

Output: **Service Components**

```
{
NumOfCom=1
Add S_1 to C_1
For (j=2 to nos){
        For (i=1 to NumOfCom) do{
        If ( T(C_i , S_j)>= α ){
        add S_j to Ci;
        break;
                }
        }
If (S_j is not yet added to any service components) then{
        numOfComp++;
        add S_j to S_{nomOfComp};
        }
}
```

Figure 2 Component Identification Algorithm

Extract Class Diagram

Input**: A Component in Clustered CRUD Matrix**

Output: **Class Diagram**

```
{
NumOfClasses = 1
For every Cluster from BE_l to BE_m & Method_k to Method_n{
        Add BE_l to Class_1
        Add BE_l to Class_1
        For (j= ( l+1) to m ) {
                For (i = 1 to numOfClasses ) do {
                        If (Coupling (Class_i , BE_j) > K) {
                        Add BE_j to Class_i
                        Break;
                        }//if
                }//for
                If (BE_j) is not yet Added to any Class then{
                numOfClass++;
                Add BE_j to Class_{numOfClass}
                }//if
        }//for
Define Intensity array with size number of Classes
        For ( i = k to n ) // number of methods{
                For (j = 1 to numOfClasses ) {
                        For (K = 1 to NumOfBEsInTheClass_i){
                        Intensity_j+ = CRUD(M_I , BE_K)
                        }//for
                }//for
        M= finde the Maximum Intensity
        Add M_I to Class_m
        }//for
        For each occurance of "C" in the Matrix in Cell " Row = row , Column = col"{
                For (j=l to m) { // Number Of Bes
                        If(CRUD(Method_row , BE_j) # {} && j#col ){
        Relation(ClassOf(BE_j) , ClassOf(BE_col)) = 1
                        }//if
                }//for
        }//for
}//for
```

Figure 3 Class Diagram Extraction Algorithm

4. Evaluation

This section presents evaluation of ASRM by applying it on a real case study and compares it with real expert's outputs. After having prepared the CRUD matrix, semantic relationships should replace by the proper weight: 1 for "C", 0.75 for "U", 0.5 for "D" and 0.25 for "R". The matrix is then clustered and services are identified through ASIM and ASSM methods.

The case study is for modeling a library system; Figure 4 shows a CRUD Matrix of a library system:

EBP \ BE	Reservation	Patron	Book Title	Book Copy	Loan	Book Copy On Loan
Reserve Book	C	R	R			
View Reservation	S1 R					
Search For Book Title			R	R		
Correct Reservation	U					
Add Patron		C				
Update Patron Information		U				
Enter New Book Information		S2	C	C		
View Patron Information		R				
Delete Book Copy Information				D		
Remove Book From Library			D			
Update Book Information			U	U		
Remove Old Loan Information					D	D
Check out books	U	R		U	S3 C	C
Check In books		R		U	U	U
Print Loaned Book reports					R	R
Print Book Title			R			
View Overdue Books						R
Remove Old Reservation Information	D					

Figure 4 Library System CRUD Matrix

	S1	S2	S3
S1		0.75	1.25
S2			1.5
S3			

Figure 5 amount of semantic relationship between components

Realized components: Figure 5 shows adjacency matrix indicating total relationship intensity between services. This will be used to identify the components.

Error! Reference source not found. shows an adjacency matrix indicating the coupling between classes. There are two BEs (Loan and Book copy on loan) which have maximum coupling and can be realized in one class. After identifying and specifying of the classes and the relationship between them, you can see the resulting class diagram based on ASRM in Figure 6. The second diagram has been drawn by domain experts just to compare with the ASRM diagram in order to see the applicability of ASRM method (see Figure 7).

Table 1 coupling between BEs

	Reservation	Patron	Book Title	Book Copy	Loan	Book Copy On Loan
Reservation		0.36	0.18	0.20	0.3	0.29
Patron			0.07	0.27	0.44	0.44
Book Title				0.58	0	0
Book Copy					0.53	0.53
Loan						0.95
Book Copy On Loan						

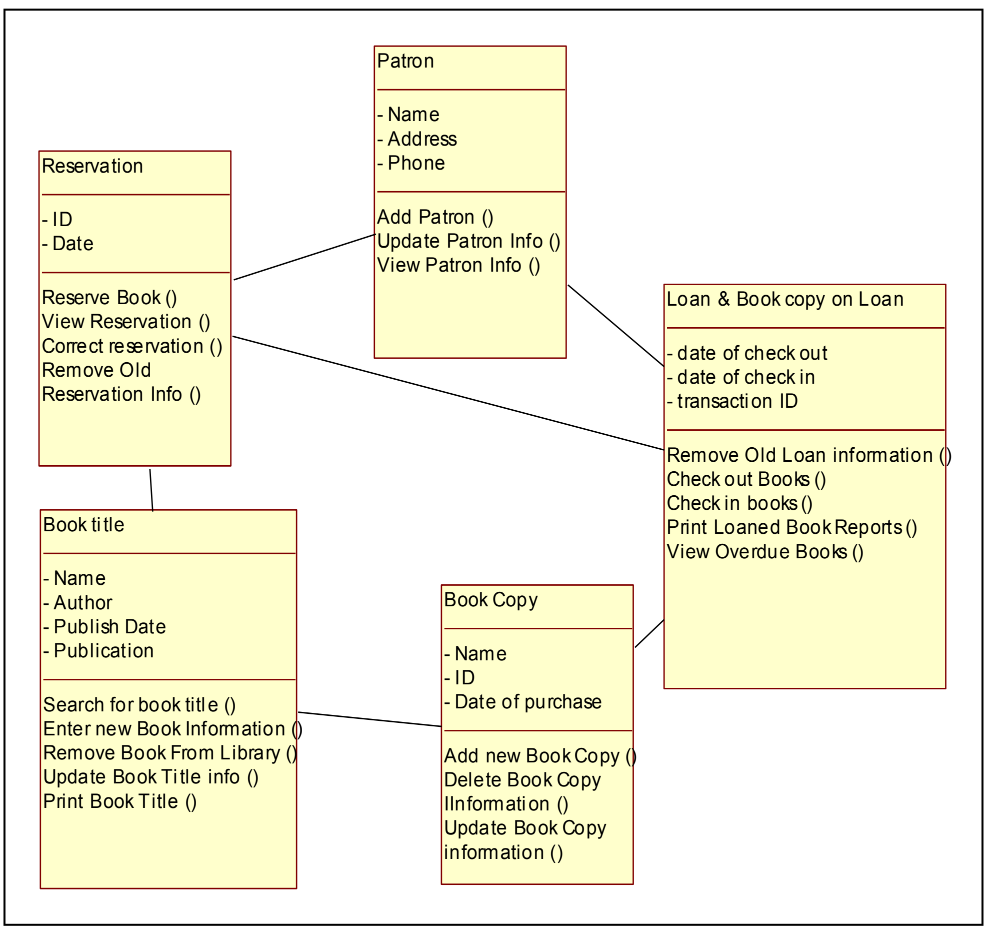

Figure 6 Library class Diagram based on ASRM approach

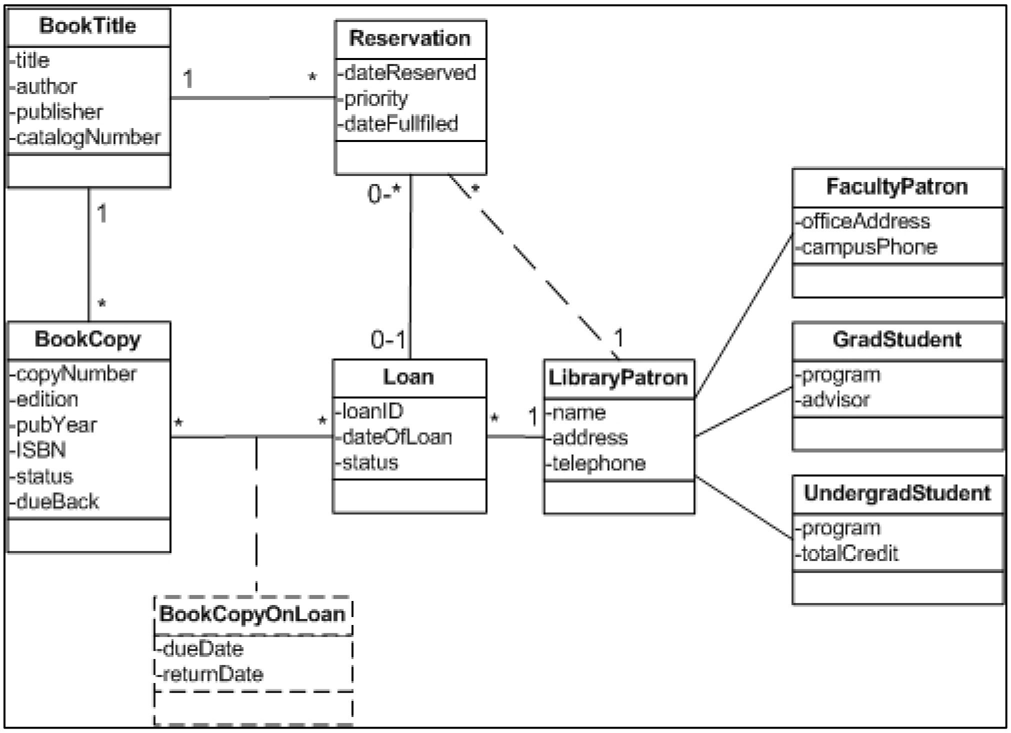

Figure 7 Expert's Class Diagram for Library System

The comparison of the two diagrams can be categorized in four items:

- ✓ class identification: very similar
- ✓ method assignment: similar
- ✓ semantic relationship between classes: except book copy and loan they are similar
- ✓ number of components: similar

The verification of the method was tested by using the diagram to build a real library system.

5. Conclusion

This paper proposes a novel method to fully realize service model by using components and object oriented world. The proposed method called ASRM consists of two major steps: component identification and class diagram extraction. And one input: Clustered CRUD matrix, which would be automatically transformed to component and class model. The aim of this method is to reduce the need of human based perspective guidelines and propose a general method, which not be specified for a single domain.

Refinement of the ASRM by adding additional functionalities such as inheritance to class diagram and also applying grasp patterns is considered as future work as well as developing an integrated toolset to effectively implement these methods.

Acknowledgments

The project was partially funded by Shahid Beheshti University under the program Automated Software Engineering Research Group, Faculty of Electrical and Computer Engineering.

References

[1]. B. Portier, "SOA terminology Overview, Part 3: Analysis and design," 2007. https://www.ibm.com/developerworks/webservices/library/ws-soa-term3/

[2]. A. Arsanjani, S. Ghosh, A. Allam, T. Abdollah, S. Gariapathy, and K. Holley, "SOMA: a method for developing service-oriented solutions," IBM Syst. J., vol. 47, no. 3, 2008, pp. 377-396.

[3]. U. Wahli, L. Ackerman, G. Hodgkinson, and L. Olson, "Buliding SOA Solutions Using the Rational SDP," IBM Redbook, 2007

[4]. j. Amsden, "Modeling SOA: Part3. Service Realization " October 2007; http://www.ibm.com/developerworks/rational/library/07/1016_amsden/

[5]. Jamshidi, P., Khoshnevis, S., Teimourzadegan, R., Nikravesh, A., Khoshkbarforoushha, A., and Shams, F.: 'ASSM: Toward an automated method for service specification' Services Computing Conference IEEE Asia-Pacific, 2009, pp. 451-456

[6]. A. Dan, R. Johnson, and A. Arsanjani, "Information as a Service: Modeling and Realization," Systems Development in SOA Environments, 2007. SDSOA '07: ICSE Workshops 2007. International Workshop on, pp. 20-26.

[7]. Zhang, L.J., Zhou, N., Chee, Y.M., Jalaldeen, Ponnalagu, K., Sindhgatta, R.R., Arsanjani, A.,

Bernardini, F.: 'SOMA-ME: A platform for the model-driven design of SOA solutions', IBM Systems Journal, 2008, volum. 47, no. 3, pp. 397-413
[8]. T. Erl, Service-Oriented Architecture: Concepts, Technology, and Design, Prentice Hall PTR , 2005
[9]. M.P.Papazoglou, W.J.Houvel, "Service Oriented design and development methodology," Int. J. Web Eng. Technol., vol. 2, no. 4, 2006, pp. 412-442.
[10]. V. De Castro, E. Marcos , and J.M Vara, "Applying CIM-to-PIM model transformations for the service-oriented development of information systems," Inf. Softw. Technol., vol. 53, no. 1, pp. 87-105. 2010;
[11]. S. Khoshnevis, P. Jamshidi, R. Teimourzadegan, A. Nikravesh, A. Khoshkbarforoushha, F. Shams, "ASMEM: A Method for AutomatingModel Evolution of Service-Oriented Systems", Maintenance and Evolution of Service-Oriented Systems (MESOA) 2009, Co-located with the 25th IEEE ICSM 2009, Edmonton, Canada.
[12]. X.F.Xu , Z.J. Wang, and D.C. Zhan, "A Survey of Business Component Identification Methods and Related Techniques,," International Journal of Information Technology, vol. 2, no. 4, 2006, pp. 513-613.
[13]. M. Ibrahim, and R. Ahmad "Class Diagram Extraction from Textual Requirements Using Natural Language Processing (NLP) Techniques" Computer Research and Development, 2010 Second International Conference on, pp. 200-204.
[14]. H. Krishnan, and P. Samuel, "Relative Extraction Methodology for class diagram generation using dependency graph," Communication Control and Computing Technologies (ICCCCT), 2011 IEEE International Conference on, pp. 815-820.