A Value-Oriented Iterative Service Modeling Process

Xiaofei Xu, Chao Ma, Zhongjie Wang

Research Center of Intelligent Computing for Enterprises and Services (ICES),
School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
xiaofei@hit.edu.cn, machao8396@163.com, rainy@hit.edu.cn

Abstract. In order to help service designers build high-quality service models quickly and accurately, this paper presents a value-oriented iterative service modeling process. This modeling process combines service business perspective and service value perspective together, and it is executed interactively between service model space and service value space, simultaneously, some value-awareness activities (e.g. value proposition, value analysis, value optimization, etc) are executed iteratively among different layers in the whole model process, so as to make service models can support the implementation of the expected values to a great extent. At last the rationality of the modeling process is explored.

Keywords: business perspective; value perspective; value-awareness; expected values; rationality.

1   Introduction

In recent years modern service industry was developing fast, and it has become the new economic growth point. The accurate execution of service business needs to be supported by service systems. Service engineering methodology is used to help service designers and service developers define, describe, design, optimize and implement the high-quality service systems accurately and quickly, which is viewed as a focus of research in service engineering field [1]. Service modeling process is an important part of service engineering methodology [2]. Service modeling process is used to carry on requirement analysis, business design, model transformation and mapping for service systems.

IBM SSME research group believes "service is the interactive process and behaviors between service providers and customers to create and share value" [3]. According this, various tangible and intangible values are regarded as the output of service systems [4]. Service values are an important characteristic of service systems. Service systems can be gradually transformed from service models by utilizing model-driven approach, so the quality of service models determines the quality of the service systems. The sole criterion for evaluating the quality of a service model is whether and to what degree the model can support the implementation of the values that are expected by both customers and providers [5]. Therefore, it is necessary to

carry on value-awareness activities in the whole service modeling process, so as to obtain a high-quality model.

Value-awareness means that "the value is regard as the basis of the service model design and decision-making in each phase of service life cycle, and before moving to the next phase, the value-oriented service model analysis must be executed to insure the implementation of all the expected values". In service modeling process, the corresponding value-awareness activities include value proposition, value expressing, value keeping, value decomposition, value analysis and value optimization. In order to better carry on value-awareness activities, this paper presents a value-oriented iterative service modeling process.

In this modeling process, not only service functional modeling from business perspective is needed, but also service value modeling from value perspective is needed, and this modeling process is executed interactively between service model space and service value space to insure the consistency in service models and value models. Service systems may refer to various stakeholders. At the beginning of the modeling process, these stakeholders understand the service systems partially and simply, therefore value-oriented model analysis, model optimization and feedback need to be executed iteratively to modify and deepen the understanding of service system incessantly.

The remainder of this paper unfolds in the following manner. Section 2 introduces service modeling theory. Section 3 presents a value-oriented iterative service modeling process. And then the rationality of the modeling method is explored in section 4, followed by conclusions in the last section.

2 Service modeling theory

For now, there are lots of study results of service modeling method from different perspectives (e.g. traditional software engineering, model-driven, domain analysis, semantics-driven).

From the traditional software engineering perspectives, the typical methods include SOMA [6], SODA [7] and SOUP [8]. By means of some methods such as domain decomposition, target service modeling, SOMA built service models, and then identified a set of self-existent service. And aiming at each service, some aspects (e.g. component specification, service interface specification, information, etc.) of the service are designed. At last a specific technique is selected to implement the services and combine these services tighter to realize a service system according to the layered structure. SODA can carry on various analysis and verification for service models, and support the flexible reuse, the reverse engineering, the automatic code generation, and so on. SOUP is a software method that is used to built and manage SOA project by utilizing RUP and XP. RUP is used to initially build the new service systems, and XP is used to analyze, assemble, reuse and optimize the existing service systems.

From the model-driven perspectives, the typical methods include the model driven service engineering process [9], the service model driven architecture (SDMA) [10]. In the model driven service engineering process, the requirement model, the enterprise model, the execution model and the IT management model are respectively

built for service systems. The service system is composed of the high level goals, the low level service components and the sensors embedded during runtime. Various phases are connected by utilizing model-driven approach. In SMDA, the service requirement model, the service behavior/capability model and the service execution model are respectively built for service systems. By means of model-driven approach, the three models are gradually built layer by layer, and the whole modeling process is top-down.

From the domain analysis perspectives, the typical methods include the SOA domain analysis method [11], the framework-based IT service engineering [12] and the rule-based configuration approach [13]. In the SOA domain analysis method, the VINCA business model is built and instantiated from top to down by utilizing domain analysis approach, simultaneously, the web services are virtualized from bottom to up, at last in the middle the VINCA business model and the web services are combined together. In the framework-based IT service engineering, service templates are used as a model specification to describe service. In rule-based configuration approach, a configurative service system is built to meet the requirements of the group customers by embedding a set of variable parameters into service models and service systems. On this basis, the individual requirements of each customer are taken into consideration, and the concrete values of various parameters in service systems are confirmed, at last the service systems that can meet service requirements are built by utilizing the automatic configuration in combination with manual configuration.

From the semantics-driven perspectives, the agent-based service system design method (Tropos) is a typical method [14]. In Tropos, the semantics of service business requirements (e.g. knowledge, rules) are defined by means of formal methods, and the requirements are mapped into the concrete service design and service orchestration by carrying on semantics transformation, at last the corresponding service systems is built.

Besides, Gordijn and others presented a multi-perspective service modeling method [15] [16] [17]. The method refers to three perspectives: a value perspective, a process perspective, and an information systems perspective. The first two perspectives belong to the requirements analysis and business design phase. The last one belongs to the systems implementation phase. During service modeling, firstly, a value network is built from the value perspective, and it is used to analyze the future economic profit of service business. And then from the process perspective the service tasks are decomposed from top to down and those tasks are described by utilizing UML-like language, at last the service process model is obtained. MDA approach is used to build service systems.

In the above methods, the most ones are the multi-perspective or hierarchical modeling methods. They only focus on the traditional functional and quality characteristic of service systems, and ignore the value characteristic. But the value is the most characteristic of service systems. The method presented by Gordijn takes the value characteristic into consideration. It carrying on the economic values calculation in the strategy layer, but it did not take the value as the constraint for the whole service modeling. The value-oriented iterative service modeling process can solve the above problem. It has the three advantages: 1) it focus on value characteristic, and combines service business perspective and value perspective together; 2) it is interactive; 3) it is iterative.

3 Service modeling process

Value-oriented iterative service modeling process is shown in figure 1. From transverse look, this process carries on interactively between service model space and service value space. From fore-and-aft look, this process carries on iteratively among different layers. The goal of this modeling process is take the expected values as the constraints for service modeling, so as to make service model can support the implementation of the expected values. The quantitative analysis of the expected values is taken as the basis of policy-making of various phases in this service modeling process.
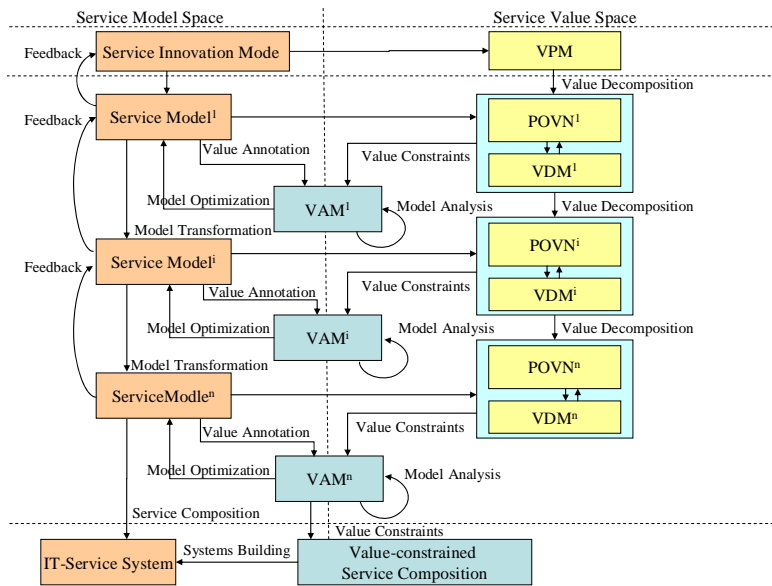

Fig. 1. Schematics of the service modeling process

3.1 Service model and value model

As is shown in figure 1, in service model space, service models are built layer by layer from coarse-grained elements into fine-grained ones. Service model is composed of service functional elements and the relationships among the various elements. It can be denoted as SM=(SE, R), where SE is service functional elements including two kinds: 1) service behaviors; 2) sub-process that is composed of a set of service behaviors. Service behaviors can be denoted as se= (PA, Fe, AO, ST, RES, QoS), where:

- PA is the set of participants of se;

- *Fe* is the farter behaviors of *se*, *Fe* is transformed into a set of fine-grained behaviors, and *se* is one of them;
- *AO* is the set of action objects which are manipulated by *se* and whose state are changed by the effect of *se*;
- *ST* is a set of *ao*'s state transitions. In $\forall st \in ST$, $\exists ao \in AO$ makes $st = ao.s_i \rightarrow ao.s_j$.
- *RES* is a set of necessary resources of *se*;
- *QoS* is a set of quality parameters that are attached to *se*.

In the beginning of service modeling process, some service business can not be directly described as a service behavior, and only can be described as a sub-process. In the subsequent service modeling process, the sub-processes are gradually refined, until all the sub-processes are transformed into the service behavior that can be implemented by service component (e.g. web service).

In service value space, the four kinds of value models are used in the iterative and interactive service modeling process. They are value proposition model (VPM), participant oriented value network (POVN), value dependency model (VDM) and value annotation model (VAM).

VPM is used to identify various end-customers and end-providers in the top service that will be designed and implemented, and identify which service values they expected to obtain from the top service.

It is denoted as VPM=(*TS*, *EPA*, *TV*), where *TS* is the top service composed of the attribute *name* and the attribute *description*, *TS* would be decomposed into a series of sequential sub-services and service behaviors during service modeling; *EPA* is the set of end-participants composed of the end-customers set *EC* and the end-providers set *EP*; *TV* is the set of the top value, $\forall v_i \in TV$, service value v_i can be denoted as v_i=(*BA*, *CA*), where:

BA is the set of basic attributes of v_i, *BA*={v_ID, *Name*, *Producer*, *Receiver*, *View*, *State*, *rc*, *rc.Type*, $rc.s^I$, $rc.s^O$, Fv_ID, *B*, *C*, *E*, *CON*}:
- v_ID, *Name* are respectively the unique identification and name of v_i;
- *Producer*, *Receiver* are respectively the producer set and the receiver set of v_i;
- *View* is the type of v_i's *Receiver*, *View* includes two types: customers *C* and service providers *P*;
- *State* is the granularity of v_i, *State* includes the root value *RV*, the interim value *IV* and the atomic value *AV*;
- *rc* is the realization carrier that v_i is attached to. It means that v_i's effect is exhibited as the improvement of *rc*.
- *rc.Type* is the type of v_i's *rc*, *rc.Type* includes *ECV*, *TIV*, *RUV* and so on;
- $rc.s^I$, $rc.s^O$ are respectively the initial state and expected final state of *rc*;
- Fv_ID is the farter behaviors of v_i, Fv_ID is decomposed into a set of fine-grained values, and v_i is one of them;
- *B*, *C*, *E* are respectively the direct benefits, the direct cost and the indirect profits of v_i;
- *CON* is the value profit constraint. $\forall q_j \in CON$, q_j can be used to evaluate the a specific characteristic of the service in which v_i's receiver is participating.

CA is the set of constraint attributes of v_i, *CA*={*e(B)*, *e(C)*, *e(E)*, *e(CON)*}, where $e(B)$={ B_{best}, $min(B)$}, B_{best} is the value of *B* during v_i's receiver accepts the best

service, $min(B)$ is the worst value of B that v_i's receiver can accept; (C), $e(E)$ are similar to $e(B)$; $\forall econ \in e(CON)$, $econ$ is the constraint set of some quality parameter q_j, $econ$ include basic constraint, comparative constraint and conditional constraint.

POVN is used to identify which enablers that are needed to import so as to realize service. POVN is also used to describe how the various values are exchanged among the participants (customers, providers, enablers).

POVN is denoted as POVN=$\{N, E\}$, where N is the node set in POVN, $\forall node \in N$, $node$ is a participant. $N=C\cup P\cup EN$, where C is the customer subset, P is the provider subset and EN is the enabler subset. E is the edge set in POVN, $\forall edge \in E$, $edge=(se, v)$, $\exists participant_i$, $participant_j \in N$, $participant_i$, $participant_j$ are two endpoints of $edge$, se is the service functional elements that $participant_i$ and $participant_j$ refer to, and v is the corresponding value of se.

VDM is used to describe the relationships among the various edges in POVN. The relationships are the dependency relationships among the various service values in POVN. VDM is denoted as VDM = (N, SN, E^D, E^I), where:

- $N=\{node_1, node_2,\ldots, node_n\}$, $\forall node_i \in N$, $\exists edge_i \in$ POVN.E, $node_i=edge_i =(se_j, v_i)$;
- $SN=\{super\_node_1, super\_node_2,\ldots, super\_node_m\}$, $super\_node_k \subseteq N$, $|super\_node_k|>1$, where $super\_node_k$ is a set of edges in POVN, and its constitutive rules: 1) to take the set of edges $\{(se_j, v_p),\ldots, (se_l, v_q)\}$ as $super\_node_k$, and the participants of any two edges are same; 2) to take the set of edges $\{(se_j, v_p),\ldots, (se_j, v_q)\}$ as $super\_node_k$, and the service functional element of any two edges is same;
- $\forall e^D \in E^D$, e^D is a directed super edge, it is denoted as $e^D=(a(e^D), c(e^D))$, where $a(e^D)$, $c(e^D)$ may be the super node, and may also be node. $E^D = CDE \cup ADE \cup SDE \cup TSDE$, where CDE is the set of composition dependency edges, ADE is the set of aggregation dependency edges, SDE is the set of support dependency edges, $TSDE$ is the set of timing sequence dependency edges;
- $\forall e^I \in E^I$, e^I is an undirected super edge, $e^I \subseteq N$, $|e^I|>1$. $E^I = DPDE \cup SPDE$, where $\forall d \in DPDE$, $d=DiffProducer(v_1,v_2,\ldots,v_n)$, $\forall d \in SPDE$, $d=SameProducer(v_1,v_2,\ldots,v_n)$, and let $pset=\{v_1,v_2,\ldots,v_n\}$.

VAM is used to describe the corresponding relationships between service values and service functional elements. It is denoted as VAM= (SM, VAT, $R(se, vat)$), where VAT is the set of value annotation tables, $\forall vat_k \in VAT$, vat_k is a set of values, and their corresponding service functional element is same; $R(se, vat)$ is the set of the corresponding relationships between se and vat. $\forall v_i \in vat_k$, $v_i= (BA, CA, DA)$, where DA is the set of impendency attributes of v_i, $DA= E^D\cup E^I$.

For more details of value models please refer to the references [18][19].

3.2 Detailed modeling process

The detailed value-oriented iterative service modeling process is shown in figure 2. This modeling process include 13 phases, and each phase is composed of serval steps.
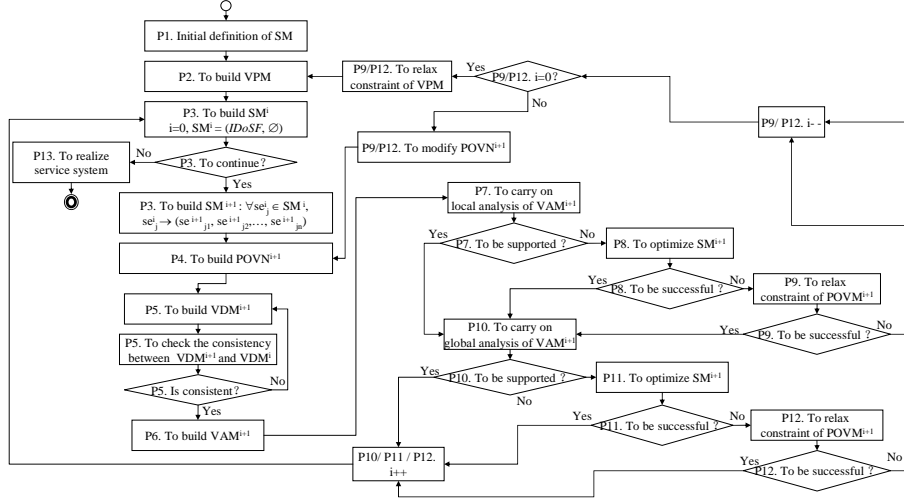
Fig. 2. Detailed modeling process

The detailed modeling process is as follows:

Phase1: To give initial definition of service model, denoted as *IDoSM*= (*IDobss*, *IDoSP*, *IDoSV*, *IDoSF*);

Step1. To initialize *IDobss*, *IDobss* is the business scope of the service;

Step2. To initialize *IDoSP*, *IDoSP* is the service participant set in the service;

Step3. To initialize *IDoSV*, *IDoSV* is the service value set provided by the various service participants in *IDoSP*;

Step4. To initialize *IDoSF*, *IDoSF* is the service function set supporting the realization of *IDobss*;

Phase2: To build VPM, denoted as VPM= (*TS*, *EPA*, *TV*);

Step1. To let *TS.name* = *IDobss*, and assign values to *TS.description*;

Step2. To identify end-customer set *EC* and end-provider set *EP* from *IDoSP*, and let *EPA*= $EC \cup EP$;

Step3. To initialize *TV*=$\varnothing$, n=|*EC*|+|*EP*|, and add *n* service values into *TV*;

Step4. $\forall c_i \in EC$, to assign values to *BA* of service value v_i that end-customer c_i expects to obtain according to *IDoSV* and the domain knowledge;

Step5. $\forall p_j \in EP$, to assign values to *BA* of service value v_j that end-provider p_j expects to obtain according to *IDoSV* and the domain knowledge;

Step6. $\forall c_i \in EC$, end-customer c_i would consult with the corresponding end-providers, and then to assign values to *CA* of service value v_i ;

Step7. $\forall p_j \in EP$, end-provider p_j would consult with the corresponding end-customers, and then to assign values to *CA* of service value v_j;

Phase3: To build service model, denoted as SM^i= (*SE*, *R*); SM^i would be gradually transformed layer by layer by means of MDA;

Step1. If i=0, to initialize SM^i= (*SE*, *R*), let SM^i= (*IDoSF*,$\varnothing$);

Step2. To judge whether or not all the service elements se^i_j belonging to SM^i.*SE* can be realized by service components (e.g. web service);

Step3. If it can be, then go to Phase13;

Step4. $\forall se^i_j \in$ SM^i.SE, if se^i_j can not be realized, to decompose se^i_j into (se^{i+1}_{j1}, $se^{i+1}_{j2}, \ldots, se^{i+1}_{jn}$), and then SM^i is transformed into SM^{i+1};

Phase4: To build POVN^{i+1} corresponding to SM^{i+1}, denoted as POVN^{i+1}= (N, E);

Step1. $\forall se^{i+1}_j \in$ SM^{i+1}.SE, to identify the customer set C, the provider set P and the enabler set EN from se^{i+1}_j.PA, and to let POVN^{i+1}.N= C $\cup$P $\cup$ EN;

Step2. $\forall (se^i_j, v^i_h) \in$POVN^i.E (if i=0, $\forall v^i_h \in$VPM.TV), according to the decomposition relationships between se^i_j and the corresponding service elements of SM^{i+1}, by utilizing some decomposition method of service value, to decompose v^i_h into v^{i+1}_{h1}, $v^{i+1}_{h2}, \ldots, v^{i+1}_{hm}$, and then to identify all the corresponding relationships ($se^{i+1}_{jk}, v^{i+1}_{hk}$), and add ($se^{i+1}_{jk}, v^{i+1}_{hk}$) into POVN^{i+1}.E;

Step3. $\forall (se^{i+1}_j, v^{i+1}_h) \in$POVN^{i+1}.E, according to se^{i+1}_j, to assign values to BA of service value v^{i+1}_h;

Step4. $\forall (se^{i+1}_j, v^{i+1}_h) \in$POVN^{i+1}.E, according to v^{i+1}_h.Fv, to divide all the service values of POVN^{i+1} into a group of subsets {$subV_1$, $subV_2, \ldots, subV_n$}, Fv of any two service values in each subset is same;

Step5. $\forall subV_k \in$ {$subV_1$, $subV_2, \ldots, subV_n$}, $\forall v^{i+1}_h \in subV_k$, according to CA of v^{i+1}_h.Fv, by utilizing some decomposition method of service values constraint, to assign values to CA of v^{i+1}_h;

Phase5: To build VDM^{i+1} corresponding to POVN^{i+1}, denoted as VDM^{i+1}= (N, SN, E^D, E^I);

Step1. $\forall edge^{i+1}_h \in$POVN^{i+1}.E, to let VDM^{i+1}.node$^{i+1}_h$= $edge^{i+1}_h$=(se^{i+1}_j, v^{i+1}_h);

Step2. $\forall subE \subset$POVN^{i+1}.E, to let VDM^{i+1}.super_node$^{i+1}_k$=subE, and subE =$\{(se^{i+1}_j, v^{i+1}_p), \ldots, (se^{i+1}_j, v^{i+1}_q)\}$, or subE=$\{(se^{i+1}_j, v^{i+1}_p), \ldots, (se^{i+1}_l, v^{i+1}_q)\}$;

Step3. If i=0, to skip this step; $\forall node^{i+1}_h \in$VDM^{i+1}.N, according to v^{i+1}_h.Fv, to identify CDE and ADE, and then add CDE and ADE into E^D;

Step4. To identify SDE, TSDE, SPDE and DPDE between various service values of VDM^{i+1}, and to add SDE, TSDE into E^D, to add SPDE, DPDE into E^I;

Step5. To judge whether or not all the consistency between SDE, TSDE, SPDE, DPDE of VDM^{i+1}and the corresponding ones of VDM^i is maintained;

Step6. If it is, then go to Phase6;

Step7. If it is not, then go back to Phase5_Step 4;

Phase6: To build VAM^{i+1}, denoted as VAM^{i+1}= (SM^{i+1}, VAT, $R(se^{i+1}_j, vat)$);

Step1. $\forall (se^{i+1}_j, v^{i+1}_h) \in$POVN^{i+1}.E, according to the corresponding relationship between v^{i+1}_h and se^{i+1}_j, to divide all the service values of POVN^{i+1} into a group of subsets {$subV_1$, $subV_2, \ldots, subV_m$}, the corresponding service element of any two service values in each subset is same; each subset is related to one vat;

Step2. $\forall subV_k \in$ {$subV_1$, $subV_2, \ldots, subV_m$}, $\forall v^{i+1}_h \in subV_k$, $\forall d \in SDE \cup TSDE$, if $v^{i+1}_h \in cons(d)$, then add d into v^{i+1}_h.DA; $\forall d \in SPDE \cup DPDE$, if $v^{i+1}_h \in pset(d)$, then add d into v^{i+1}_h.DA; $\forall d \in CDE \cup ADE$, if $v^{i+1}_h \in ante(d)$, then add d into v^{i+1}_h.DA;

Phase7: To carry on the local analysis of VAM^{i+1} [20];

Step1. $\forall v^{i+1}_h \in$VAM^{i+1}, to calculate v^{i+1}_h based on the Qos of the corresponding service functional elements;

Step2. To judge whether or not the implementation of all the values can meet the constraints;

Step3. If there are the constraints of some values (represented by *UnmetVset*) that can not be met, then go to Phase8;

Step4. If the constraints of all the values can be met, then go to Phase10;

Phase8: To optimize SM^{i+1};

Step1. To modify the *QoS* of the corresponding service functional elements of the values belonging to *UnmetVset* by utilizing some heuristic optimization method;

Step2. If the gaps between the implementation and the constraints of the values belonging to *UnmetVset* can be eliminated, then go to Phase10;

Step3. If the gaps can not be, then go to Phase9;

Phase9: To eliminate the gaps by relaxing the constraints of the values of $POVN^{i+1}$;

Step1. To relax the constraints of the values belonging to *UnmetVset*;

Step2. To judge whether or not the gaps can be eliminated;

Step3. If the gaps can be, then go to Phase10;

Step4. If the gaps can not be, then let $i=i$-1:

Step4.1. If $i==0$, to relax the constraints of the values in VPM which are the father value of the values belonging to *UnmetVset*, then go back to phase2;

Step4.2. If $i\neq0$, to relax the constraints of the values in $POVN^{i+1}$ which are the father value of the values belonging to *UnmetVset*, and then go back to phase4;

Phase10: To carry on the global analysis of VAM^{i+1};

Step1. FV^{i+1} is the set of the *Fv* of all the values in VAM^{i+1}, $\forall v^i_k \in FV^{i+1}$, to calculate all the values of FV^{i+1};

Step2. To judge whether or not the implementation of all the values can meet the constraints;

Step3. If there are the constraints of some values (represented by *UnmetFVset*) that can not be met, then go to Phase12;

Step4. If the constraints of all the values can be met, then $i=i$+1, and go to Phase3;

Phase11: To optimize SM^{i+1};

Step1. To modify the *QoS* of the corresponding service functional elements of the values decomposed from the values belonging to *UnmetFVset* by utilizing some heuristic optimization method;

Step2. If the gaps between the implementation and the constraints of the values belonging to *UnmetFVset* can be eliminated, then $i=i$+1, and go to Phase3;

Step3. If the gaps can not be, then go to Phase12;

Phase12: To eliminate the gaps by relaxing the constraints of the values of $POVN^{i+1}$;

Step1. To relax the constraints of the values decomposed from the values belonging to *UnmetFVset*;

Step2. To judge whether or not the gaps can be eliminated;

Step3. If the gaps can be, then $i=i$+1, and go to Phase3;

Step4. If the gaps can not be, then let $i=i$-1;

Step4.1. If $i==0$, to relax the constraints of the values belonging to *UnmetFVset* in VPM, and then go back to phase2;

Step4.2. If $i\neq0$, to relax the constraints of the values belonging to *UnmetFVset* in $POVN^{i+1}$, and then go back to phase4;

Phase13: To realize service system by utilizing service composition approach;

Step1: To carry on the quantitative evaluation on the value-oriented priority of service functional elements in SM^{n} [21];

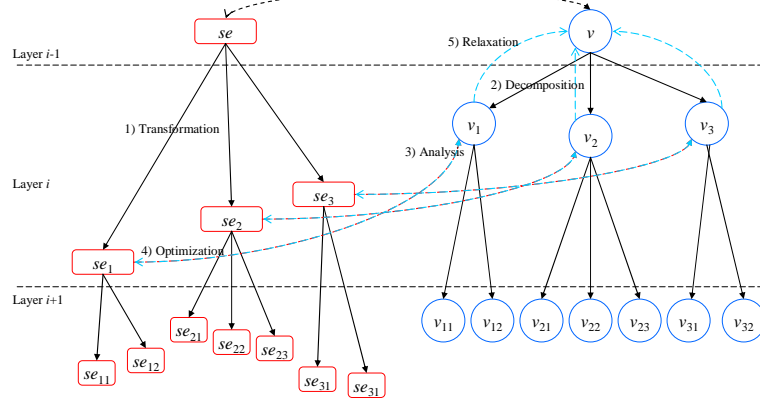Step2: To carry on VOP based service composition.

Fig 3. An example of the modeling process

In order to better explain the above modeling process, a simple example is given in figure 3. It is assumed that there is a service functional elements se in the layer i-1, its corresponding value is v, and the implementation of v can meet v's own constraints. In the layer i, firstly, se is transformed into $<se_1, se_2, se_3>$; secondly, v is decomposed into $\{v_1, v_2, v_3\}$, and there is a corresponding relationship between se_1 and v_1; the next phase is to analyze whether or not the constraints of v_1 can be met by v_1's implementation that is under the support of se_1. If the constraints of v_1 can not be met, the next phase is optimize se_1 by modifying its QoS; If the gaps between v_1's implementation and v_1's constraints can not be eliminated by optimizing se_1 or by relaxing v_1's own constraints, the next phase is to go back to the layer i-1 to relax v's constraints. If all the constraints of se_1, se_2 and se_3 can be met, then go to the layer i+1.

4   Rationality of modeling process

The iteration and interactive are two key characteristics of the modeling process. The rationality of them is respectively discussed below.

There are various stakeholders in service system, their requirements are different and their degree of the understanding for service system is also different. Each stakeholder only can understand service system from their own point of view, and then pursue their own goal. Therefore, the stakeholder that can understand service system from a global perspective is impossible to exist. During service system designing, it is necessary to interactively carry on modeling process among various stakeholders, and to combine those understandings for service system together, so as to make service system can meet the requirements of all the stakeholders. As is shown in figure 4, the different stakeholders have different perspectives, and their range of the cognition for service and service system is different, but there are the intersections

among various cognitive ranges, it is rational to interactively carry on modeling process among various stakeholders.
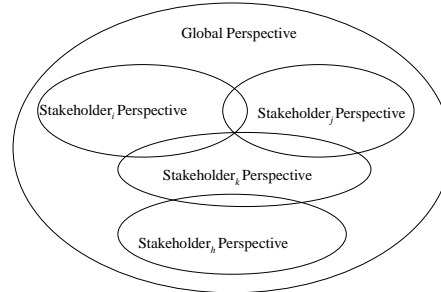

Fig 4. Different perspectives of the different stakeholders

In this paper, the modeling process includes two perspectives: values perspective and business perspective. The modeling process is executed interactively between service model space and service value space. From business perspective, service functional model is gradually built; the model designers only focus on the functional and quality requirements, and ignore the expected values. From value perspective, several corresponding value models is built along with service functional model, the value designers mainly focus on the expected values, and did not understand much service business. But the value designers and the model designers can know what relationship between the expected values and service functional elements. Therefore, it is rational to interactively carry on modeling process between the value designers and the model designers.

The people's cognition for new things is progressive. The same goes for the cognition of the stakeholders for service systems. At the beginning of modeling, the stakeholders' degree of the understanding for service system is not high because their perspectives are local and the information obtained is few. As the stakeholders obtain some new information from others, their degree of the understanding for service system become higher, they may found some previous understanding for service system is wrong, so it is necessary to go back to adjust. It is similar to the backtracking method in software engineering.

In this paper, at the beginning of modeling, the cognition of the value designers and the model designers for service systems is limited. The cognition is constantly deepened by carrying on iteratively modeling process. For example, the model designers built SM^i, but it is founded that SM^i can not support the sufficient implementation of the expected values by model analysis, so it is necessary to go back to modify SM^i. For another example, the value designers set the constraints of the expected values in $POVN^i$, but the gaps can not be eliminated by utilizing model optimization or relaxing the constraints of the expected values in $POVN^i$, so it is necessary to go back to relax the constraints of the corresponding values in $POVN^{i-1}$. Therefore, it is rational to iteratively carry on modeling process among different layers.

5   Conclusion

This paper presents a value-oriented iterative service modeling process. On the one hand, it combines the service business perspective and service value perspective together. The expected values are taken as constraints for service models. The value characteristic of service system is focused on. On the other hand, it makes service models can support the implementation of the expected values to a great extent by carrying on interactively between service model space and service value space and by carrying on iteratively among different layers. The modeling process can be used to help service designers to build the high-quality service model, and the service system that is built based on this service model can deliver the service values to providers and customers to a great extent.

Acknowledgment

References

1. Papazoglou, M., Traverso, P., Dustdar, S., F. Leymann, Krämer, B.: Service-Oriented Computing Research Roadmap, March, http://infolab.uvt.nl/pub/papazogloump-2006-96.pdf (2006)
2. XiaoFei Xu, ZhongJie Wang, Tong Mo: Methodology for service engineering. Computer Integrated Manufacturing Systems, 13(8): 1457- 1464 (2007)
3. Jim, S., Paul, M., John B., and Daniel G.: Steps toward a Science of Service Systems, IEEE Computer, 40 (1): 71-77 (2007)
4. Chao Ma, Xiaofei Xu, Zhongjie Wang: Preliminary Discussions on Several Characteristics of Service Value. International Journal of Service Science, Management, Engineering and Technology, 1(3): 50-62 (2010)
5. XiaoFei Xu, ZhongJie Wang,: Value-Aware Service Model Driven Architecture and Methodology. IFIP: E-Government ICT Professionalism and Competences Service Science. New York, USA: Springer, 280: 277-286 (2008)
6. Arsanjani, A., Ghosh, S., Allam, A., et al: SOMA: A method for developing service-oriented solutions. IBM System Journal, 47(3): 377-396 (2008)
7. Nigam, S.: Service Oriented Development of Applications (SODA) in Sybase Workspace. Sybase Inc. whitepaper, Sybase Inc., http://www.sybase.com/detail?id=1036164 (2005)
8. Mittal,   K.:   Service   Oriented   Unified   Process   (SOUP). http://www.kunalmittal.com/html/soup.html
9. Anaby-Tavor, A., Amid, D., Sela, A., et al: Towards a Model Driven Service Engineering Process. IEEE Congress on Services 2008 - Part I, July, 503-510 (2008)
10. XiaoFei Xu, Tong Mo, ZhongJie Wang: SMDA: A Service Model Driven Architecture. Proceedings of the 3rd International Conference on Interoperability for Enterprise Software and Applications. Mar. 28-30, Madeira Island, Portugal. Enterprise Interoperability II: New Challenges and Approaches. Ricardo J.Goncalves, Jorg P. Muller, Kai Mertins, Martin Zelm (Eds). ISBN: 978-1-84628-857-9. 291-302.

11. Wang, J.W., Yu, J., Han, Y.B.: A Service Modeling Approach with Business-Level Reusability and Extensibility. IEEE International Symposium on Service-Oriented System Engineering, 23-28 (2005)
12. Specht, T., Spath, D., Weisbecker, A.: Weisbecker. Framework-based IT Service Engineering, ITI 3rd International Conference on Information and Communications Technology, 19-38 (2005)
13. Becker, J., Beverungen, D., Knackstedt, R., Matzner, M.: Configurative Service Engineering - A Rule-Based Configuration Approach for Versatile Service Processes in Corrective Maintenance. Proceedings of the 42nd Hawaii International Conference on System Sciences, January, 1-10 (2009)
14. Lau, D., Mylopoulos, J.: Designing Web Services with Tropos. International Conference on Web Services, 306-313 (2004)
15. Gordijn, J., Eck, P.v., Wieringa, R.: Requirements Engineering Techniques for e-Services. In D Georgakopoulos and M.P. Papazoglou editors, Service-Oriented Computing, The MIT Press, Cambridge (2009)
16. Gordijn, J., Yu, E., van de Raadt, B.: E-Service Design Using i* and e3value Modeling. IEEE Software, 23(3): 26-33 (2006)
17. Gordijn, J., Akkermans, H.: E3-value: Design and Evaluation of e-Business Models. In IEEE Intelligent Systems, 16(4):11-17 (2001)
18. ZhongJie Wang, XiaoFei Xu: Multi-Level Graphical Service Value Modeling Method. Computer Integrated Manufacturing Systems, 15(12): 2319–2327 (2009)
19. ZhongJie Wang, XiaoFei Xu: Service value dependency model based on layered hyper-graph. Computer Integrated Manufacturing Systems, 17(8): 1837-1846 (2011)
20. Chao Ma, Xiaofei Xu, Zhongjie Wang: A Process Algebra based Method for Value Analysis on Service Process Model. Chinese Journal of Computers, 33(11): 2177-2189 (2010)
21. Chao Ma, Xiaofei Xu, Zhongjie Wang: The Quantitative Evaluation on the Value-Oriented Priority of Service Elements. 2011 International Joint Conference on Service Science (IJCSS 2011). Taipei, Taiwan, China, May 25-27. IEEE Computer Society, 257-261 (2011)