Automatic System Modeling Approach based on Semantic Association

Peng Rong    He Keqing    Liang Peng
State Key Lab. of Software Engineering, Wuhan University
rongpeng@sklse.org   hekeqing@sklse.org

Abstract

Domain model, an important asset retrieved from domain analysis, can provide a problem-oriented solution for modeling the domain applications.

The focus of this research is to automatically modeling specific domain application according to user requirements with the support of domain models. In this paper, it proposes an association method to establish the semantic association between expected goals which is extracted from user requirement and domain models; based on the association, an automatic modeling method is put forward which utilize the modeling solution of domain models; at the end, an experimental system DGMR Manager is carried out to support the idea of this approach.

Keywords: Semantic association, model, ontology

1. Introduction

Domain engineering concerns the work required to establish a set of software artifacts that can be reused by the software engineer. The purpose of domain engineering is to identify, model, construct, catalog and disseminate a set of software artifacts that can be applied to existing and future software in a particular application domain [1]. Domain Analysis (DA) is a key activity in Domain engineering. Most domain analysis methods, such as FODA [2], RESB [3], use a specific model to play a unifying role for asset reuse across a family of systems.

Domain models can provide a problem-oriented architecture for the application domain that reflects the similarities and variations of the members of the domain [4]. How to model and create an individual target system by selecting objects from the domain model is the main topic [2, 3, 4].

In this paper, an automatic system modeling approach based on semantic association is proposed. It employs ontology based reasoning technology for automatically constructing the semantic association between domain goal models and expected goals which are extracted from user requests. At the same time, it records the usage of each domain model at runtime which can be used to measure the suitability of domain model repository and predict the evolution solution of the repository. In the end, a trial application of this approach is proposed to demonstrate its effectiveness.

2. The Framework of Approach

2.1 Ontology based Domain Goal Model Repository

As we know, user requirements always focus on describing what they want from the expected system, namely the goal of the expected system [5]. Establishing domain goal model can not only facilitate the comprehension of end users and aid the analysis of application requirement, but also benefit to manage the domain asset purposefully.

We construct the Domain Goal Model Repository (DGMR) under the guidance of the following Goal Metamodel [6] (see Figure 1). We focus on Functional Goal, since the methodological benefit of this approach can be sufficiently demonstrated on this subset of goals. What’s worth mentioning is that Functional Goal has attributes Operation and Object which are associated with Operation Ontology and Object Ontology. With the support of ontology technology [7], it possesses the capability of reasoning.

The implementation framework of DGMR is shown in Figure 2. We argue that user can describe their requirements using various format (including multimedia content), and Expected Goals (EGs) matching with user requirements can be retrieved by using various requirement analysis methods [8, 9]. The workflow of framework is as following:

(1) EGs can be modeled through Semantic Association Analysis Engine(SAAE) using the ontology and model resources in DGMR;
(2) Submit the modeling solution for user’s feedback, and store the Semantic Association (SA) record into DGMR;
(3) Get the feedback from users, and modify the SA record through SAAE;
(4) Metrics Monitor Engine (MME) monitors the changes of SAs and determine the evolution timing;
(5) Evolution Engine (EE) is triggered when the evolutionary condition is met;
(6) EE generates the recommendations on evolution solution based on SA records and submits the solution to DGMR administrator;
(7) DGMR administrator send the feedback to EE;
(8) EE realize the evolution of DGMR in the end.

The advantages of this framework include:
(1) The constructed DGMR with the support of domain ontology has better flexibility and adaptability to support the system requirement modeling;
(2) It records the using history of domain goal models in DGMA at runtime, which provides the basis for objective analysis of DGMA;
(3) The metrics criteria based on SA records can determine the proper timing for domain model evolution;
(4) All kinds of SA records can support the decision of domain model evolution plan with the support of metrics criteria.

In this paper, we focus on the definition and classification of semantic association, and the algorithms adopted by SAAE.

2.2 Related Assumptions
Three assumptions are assigned for this approach: (1) the DGMR, including domain ontology and domain models, is readily available or constructed by domain experts and requirement engineers using existing
domain engineering methods; (2) the name of Domain Functional Goal model (FG) is composed of Operation Ontology and Object Ontology, e.g. FG ManageOrder whose Operation Ontology is Manage and Object Ontology is Order; (3) the EG, which is also expressed by verb-object structure, can be retrieved from users’ requirements using exiting requirement elicitation and analysis method.

3. Semantic Association of Domain Goal Model

3.1 Expected Goal of Requirement

Users can describe their requirements using their familiar methods, including SORL [15], pseudo natural language, UML visualization modeling language etc. The goal which can satisfy perfectly the user goals is named as Expected Goal (EG), which can be acquired by retrieving the kernel predicates and objects from user requirement statements.

3.2 Domain Goal Model supported by Ontology

The domain goal model in DGMR is support by ontology technology. To a specific DG, its kernel is its Functional Goal Model (FG). Therefore, the matching from EG to DG can be regarded as matching from EG to FG. Each FG has attributes Operation and Object, which associates with Operation Ontology and Object Ontology. These domain ontologies are organized by the semantic relationships such as equivalent, transitive, inverse of, super- and sub-class of, et al., to fulfill the reasoning functionality.

The ontological knowledge and reasoning support for domain model repository is fundamental base in this paper.

3.3 Association between Expected Goal and Domain Goal Model

To construct the goal model for the system to be developed with the support of DGMR, we should establish the association between EG and DG. As Figure 3, the Association can be classified as Supported Association (SuA) and Deficient Association (DeA) depending on the way of matching, i.e. matching by SAAE or matching manually. The SuA can be classified as Direct Association (DiA) and Indirect Association (InA) depending on whether the reasoning function based on domain ontology is used or not and it can also be classified as Identical Association (IdA), Equivalent Association (EqA), Generalized Association(GeA), and Specialized Association (SpA) depending on the matching way used by SAAE. Meanwhile, the SuA can be further classified as Successful Association (SucA) and Failure Association (FaA) depending on whether the matching results can be confirmed by user or not.

![Figure 3. Classification of Association between EG and DG](image)

This classification is vital to the construction of SA, and metrics and evolution based on SA. The detailed definition of some association is presented as following:

1) **Identical Association**

IdA denotes that the EG can be directly matched with the name of DG in DGMR. An instance of IdA is shown in Figure 4a, which is also a DiA.

2) **Equivalent Association**

EqA denotes that the EG can be matched with DG’s operation ontology and object ontology in DGMR using the *equivalent relationship reasoning* of ontology. An instance of EqA is shown in Figure 4b, which is also an InA.

3) **Generalized Association**

There are three kinds of GeAs, including *Operation Generalization (OpGeA)*, i.e. the operation ontology related with EG (named as Expected Goal oPeration Ontology (EGPO) in following sections) is a subclass concept of the operation ontology in domain goal (named as Domain Goal oPeration Ontology (DGPO) in following sections), and the object ontology is equal as shown in Figure 4c; *Object Generalization (ObGeA)*, i.e. the object ontology related with Expected Goal (named as Expected Goal oBject Ontology (EGBO) in following sections) is a subclass concept of the object ontology in domain goal (named as Domain Goal oBject Ontology (DGBO) in following sections), and the operation ontology is equal as shown in Figure 4d; *Double Generalization (DbGeA)*, the operation and
object ontology in Expected Goal are both the subclass concept of the operation and object ontology in domain goal, as shown in Figure 4e. GeAssociation is obviously an InA.

4) Specialized Association

| QueryOrder | QueryOrder | (a) Identical Association |
| SearchOrder | ManageOrder | (b) Equivalent Association |
| QueryOrder | QueryBill | (c) Generalized Association-1 |
| QueryOrder | ManageBill | (d) Generalized Association-2 |
| QueryOrder | QueryBill | (e) Generalized Association-3 |
| ManageOrder | QueryOrder | (f) Specialized Association-1 |
| QueryBill | QueryOrder | (g) Specialized Association-2 |
| QueryBill | QueryOrder | (h) Specialized Association-3 |

Figure 4. Instances of Supported Association

There are three kinds of SpAs, including Operation Specialization (OpSpA), i.e., EGPO is a superclass concept of DGPO, and the object ontology is equal as shown in Figure 4f; Object Specialization (ObSpA), i.e., EGBO is a superclass concept of DGBO, and the operation ontology is equal as shown in Figure 4g; Double Specialization (DbSpA), the EGPO and EGBO are both the superclass concept of the DGPO and DGBP respectively, as shown in Figure 4h. Specialized association is obviously an indirect association.

5) Deficient Association

When the EG can’t be matched with the domain model in DGMR using the four association relationships mentioned above, the EG modeling task cannot be fulfilled automatically, and it is necessary for the requirement engineers to construct the EG model manually. It is the common sense in the context of system construction that this kind of EG can not be isolated from other EGs of the system which associate with some DGs. Thus, this kind of indirect association between EG and DG is defined as DeA. For instance, the DGMR includes DGs such as “ManageBill”, “BrowseBill”, “QueryBill”, and “SortBill”, but doesn’t include DG “PrintBill”, and the operation ontology “Print” is not defined as a subclass of operation ontology “Manage”. Then, the association between PrintBill and ManageBill can’t be identified automatically. When user notices that an additional “PrintBill” function is necessary, and constructs manually the association between EG “PrintBill” and DG “ManageBill”, as shown in Figure 5, the association between them is a DeA. And the “PrintBill” is called the Deficient Goal of “ManageBill”.

3.4 EG Modeling based on DGMR

The ultimate objective for the construction of DGMR is to support the user-specific system modeling using DGMR. The key issue to achieve this objective is how to select appropriate DG to support the EG modeling. With the association definitions above, we propose two algorithms for the automatic EG modeling. The Semantic Association Record (SAR) in the following sections includes the type and EG name of the SA, which can provide useful information for the metrics and domain model evolution afterwards.

Algorithm 1: Algorithm for the association between EG and DG

1. Search for the directly matching DG for the EG in the DGMR. If the matching DG can be found, an IdA is constructed between the EG and DG, and the count of IdA to this DG, namely |DG.IdA.Count|, increases by 1. Go to step 7.
2. Split up the verb-object structure of EG into operation ontology and object ontology;
3. Search for DG in Equivalent Association: Search for the equivalent ontology of EG’s operation ontology and object ontology respectively, and construct the equivalent goal set. Compare the goal in the equivalent goal set with the DG in DGMR one by one by name matching. If matching DG can be found, an EqA is constructed between the EG and DG, and the |DG.EqA.Count| increases by 1. Go to step 7.
4. Search for DG in Generalized Association: Search for the super ontology and equivalent ontology of super ontology for the object ontology, and equivalent goal set. Compare the goal in the equivalent goal set with the DG in DGMR one by one by name matching. If matching DG can be found, an ObGeA is constructed between the EG and DG, and |DG.ObGeA.Count| increases by 1. Go to step 7.
5. Search for DG in Generalized Association: Search for the super ontology and equivalent ontology, and operation ontology with operation ontology. Compare the goal in the equivalent goal set with the DG in DGMR. If matching DG can be found, an ObGeA is constructed between the EG and DG, and |DG.ObGeA.Count| increases by 1. Go to step 7.
6. Search for DG in Deficient Association: Search for the super ontology and equivalent ontology of super ontology, and equivalent goal set. Compare the goal in the equivalent goal set with the DG in DGMR. If matching DG can be found, an ObGeA is constructed between the EG and DG, and |DG.ObGeA.Count| increases by 1. Go to step 7.
7. Add the EG to the EG model, and the EG modeling task is completed.
and specialization processing during the DG matching operation ontology in the equivalence, generalization the object ontology is assigned higher priority than the generalization and specialization process, therefore ontology tends to lose critical semantic information in the evolution of DGMR.

modeling. The records of DeA provide a good basis for sub-goals of the EG, and it can partially support the EG low recognition degree. This DG is usually one of the specialized semanteme of the EG, and hence it has readily recognized by user. The DG retrieved by SpA is remedy the requirement analysis results. Therefore, it is normally extend the system function, and possibly as accurate as the EG; but the modeling from DG generalized the semanteme of EG. This DG may not be is no semantic loss. The DG retrieved by GeA

c) Search for the super ontology and equivalent ontology of super ontology for the operation and object ontology, and construct the double generalization goal set. Compare the goal in the generalization goal set with the DG in DGMR. If matching DG can be found, a DbGeA is constructed between the EG and DG, and |DG.DbGeA.Count| increases by 1. Go to step 7;
d) Search for the ancestor super ontology for the operation and object ontology, and repeat the steps from 4a to 4c till the matching to the top ontology, if matching goal can be found, go to step 5;
5. Search for the DG in Specialized Association: Do the same task as Step 4 except that 1) replace super ontology to sub ontology; 2) If matching DG can be found, construct a relevant SpA and increases its count by 1;
6. The EG has no automatically associated DG: If the user associates the EG with an existing goal in system goal models, and this goal is related with a DG in DGMR, then a DeA is constructed between them, and |DG.DeA.Count| increases by 1. Go to Step 8.
7. The associated DG acts as the starting point for the EG modeling, and apply the Association Algorithm 2 for EG modeling.
8. The algorithm ends up.

Algorithm Description:
1) In the perspective of user recognition, the DG retrieved by IdA and EqA is mostly welcome since there is no semantic loss. The DG retrieved by GeA generalized the semanteme of EG. This DG may not be as accurate as the EG; but the modeling from DG normally extend the system function, and possibly remedy the requirement analysis results. Therefore, it is readily recognized by user. The DG retrieved by SpA is specialized the semanteme of the EG, and hence it has low recognition degree. This DG is usually one of the sub-goals of the EG, and it can partially support the EG modeling. The records of DeA provide a good basis for the evolution of DGMR.
2) Compared with the object ontology, the operation ontology tends to lose critical semantic information in the generalization and specialization process, therefore the object ontology is assigned higher priority than operation ontology in the equivalence, generalization and specialization processing during the DG matching process, for the purpose of higher degree recognition by User.

Finding the DG matched with EG is first step of system modeling. The critical support of domain model repository lies on that engineers can use the DG decomposition and modeling solution in the DGMR to model the EG, and Association Algorithm 2 provides the detailed modeling solution.

Algorithm 2: Algorithm for EG modeling
1. Take the matching DG as the root goal of modeling. If the association between EG and DG is constructed by object specification or object generalization, then replace each appearance of DG’s object in the model with EG’s object;
2. Retrieve its related modeling elements, such as NonFunctional Goal, various Constraint to construct the basic EG model;
3. Retrieve all the sub-goals of this DG in the DGMR. Classify the sub-goals into Mandatory, Optional, Alternative and Or sub-goal group, and submit these sub-goal groups to the user as the modeling solution to the EG. If there are no any sub-goals available for this domain goal, then the user can decide whether the matching DG is successful or not. If it is successful matching goal, then go to step 8, otherwise go to step 7;
4. For the Optional sub-goal group, user confirms the selection of some sub-goals in this group;
5. For each Or and Alternative sub-goal group, stakeholders confirm the selection of at least one sub-goal in this group. If stakeholders cannot recognize the decomposition of the sub-goal groups, then go to step 7;
6. For Mandatory sub-goal group, all the sub-goals in this group are mandatory for selection. If user can recognize the decomposition, go to step 8. If user cannot recognize it, go to step 7;
7. Insert the FaA record to the DG, and record the type of failure, such as MismatchFaA, MandatoryFaA, AlternativeFaA and OrFaA, and record the name of the sub-goal which can not recognized by user. Corresponding count of failure association increases by 1. Go to step 8.
8. Increases |DG.SucA.Count| by 1, and the numbers of IdA to all the sub-goals of this DG increase by 1. Go to step 9.
9. The algorithm ends up.

Regard each sub-goals (including Mandatory, Alternative, Or, Optional goal) of domain goal as an independent domain goal, and repeat the Association Algorithm 2 until there is no sub-goals available in the DGMR.

Algorithm Description:
Association algorithm 2 mainly focuses on how to utilize the domain goal model resource to support EG’s modeling. At the same time, it completes the SARs according to user feedback, which will be the objective basis for the metrics and evolution of DGMR.

6 Trial Application

This approach is applied and validated with a demonstration in the logistics domain.

A visualized domain goal model repository management tool, DGMR Manager, is implemented based on the approach presented in this paper. Its architecture is shown in Figure 2. Since the metrics and evolution function is not the focus of this paper, we only introduce its modeling function.

DGMR Manager can support the modeling of DG and EG as shown in Figure 6 and Figure 7.

![Figure 6. Domain Goal Modeling](image)

As shown in the diagram of the domain goal model (Figure 6), the goal name (e.g. ManageOrder) is a verb-object phrase. The verb “Manage” is recorded by Operation attribute related with the operation ontology “Manage”, and the object “Order” is recorded by Object attribute related with the object ontology “Order”. The functional goal can be decomposed iteratively till the operational goal. The goals can be classified as Mandatory Goal, Alternative Goal, OR goal, or Optional goal in the goal decomposition process.

As shown in Figure 7, when the user’s EG is “ManageOrder”, the tool can find the domain goal “ManageOrder” with direct association using the association algorithm 1. In this situation, the tool recommends the modeling solution of the “ManageOrder” in the DGMR to user. The user can accept the recommendation, and the associations to the “ManageOrder” are successful associations; the user can also reject the recommendation, and the associations to the “ManageOrder” are failure associations.

![Figure 7. Expected Goal Modeling](image)

With the examples shown above, the DGMR Manager can support effectively the modeling process of EG, and get vital SARs to support the domain model metrics and evolution.

7 Related Work

In paper [3], Martin L. Griss has outlined a FeatuRSEB domain analysis process which integrates the feature modeling with RSEB. It focuses on how to construct FeatuRESB feature models and use them to index into the commonality and variability present in use case and object model. Comparing with it, there are two differences: 1) it focuses on use case model but we focus on goal model; 2) the first step of using FeatuRESB must done by reuser manually, but the matching process between EG and DG in our method is performed automatically.

In paper [11], it focuses on the improvement of recording and management of requirements which is organized as requirements tree, but our approach aims at functional goal. To support an intensive reuse of requirement models, both of us adopt the domain engineering approach, but our approach pay more attention to how to record the history of domain model usage which is critical to assist the decision of domain model update.

Although many reuse based software modeling processes are all based on domain analysis [12, 13, 14], how to record the domain model usage to assist the evolution of domain assets does not clarify.

8 Summary

The following research results have been achieved:
1) The association method for matching the domain goal in DGMR according to the expected user goal;
2) The automatic modeling method for expected user goal which utilize the modeling solution of domain model;
3) The experimental system DGMR Manager implementing the association and modeling algorithm which can support the idea of this approach.

We outline our future work in several points:
1) The metrics criteria for the adaptability of domain goal models need to be established according to the SARs of DG;
2) The evolutionary algorithm of DGMR needs to be established according to the SARs and metrics criteria;
3) The function of DGMR Manager should be enhanced and this toll should be applied and validated in wider industrial context, in order to verify the effectiveness of it.

9 Acknowledgments

This research project was supported by the National Basic Research Program of China (973) under Grant No.2006CB708302 and 2007CB310801, the National High Technology Research and Development Program of China (863) under Grant No.2006AA04Z156, the National Natural Science Foundation of China under Grant No.90604005 and 60703009.

10 References