2012 International Conference on Intelligent Systems Design and Engineering Application

Applying Case-based Reasoning Technique for Customized Management

Yao Qing, Sun Yuqing

Shandong University, Jinan, Shandong, 250100, China
yaoqing@sdu.edu.cn

Abstract—**To support agile management, enterprises would face the challenges of designing and executing flexible management model. In this paper, we present a novel Customized Management (CM) model based on given goals. First, the knowledge base is established to store the standardized scenario data involved with business goals with the help of business experts. Then based on such knowledge base, a customized management model could be created by means of Case-Based Reasoning (CBR) technique so as to satisfy the specific management requirements. We present the definition of CM, describe its creating process, and discuss the experiment result.**

Keywords-Customized management; Case-Based Reasoning (CBR); Virtual Corporation

I. INTRODUCTION

The needs of an enterprise or a company are universal—quick response and excellent services to its customers. The success of an on-demand e-business requires that management model and information technology infrastructure are improved into a new architecture[1]. To achieve the goals of managing ecommerce transactions and delivering services to businesses and individual customers rapidly, enterprises face the challenges of designing and executing Customized Management(CM) flexibly.

We developed an intelligent platform of a virtual travel agency, IPVita, as a prototype system. The system was designed as a travel service company running in e-business mode like Expedia or CTrip. It has many contracted travel services spread all over the world, and they compose a virtual corporation. Management models are customized in this platform. It consists of four functional components: (1) requirement catch, (2) MC model creation, (3) MC execution monitoring, and (4) knowledge base (KB) management. The experiential MC case repository stores successful MC model cases for reuse in the future. Figure 1 gives the outline of IPVita works.

II. CBR OVERVIEW AND RELATEDWORKS

CBR is a problem-solving approach, an AI reasoning technique proposed by Roger Schank. The basic idea of CBR is to solve new problems by comparing them with old problems, which have already been solved in the past[2]. Figure 2 shows the CBR processes covering retrieval, reuse, revise, and retain[3]. Existing problems and their solutions are stored in a database of cases called a case-base. When a new problem is presented, the CBR system tries to retrieve the most similar existing problems from the case-base. The idea is that, if two problems look similar, then the solutions to these problems are also possibly similar. The concept of case similarity measure plays a crucial role in performing these processes. In the retrieval process, cases can be structured or indexed appropriately. Existing solutions are reused or suggested. If the retrieved cases are not a close match, the solution will need to be revised or adapted to provide a new solution, which will be retained in the case-base for further use[4].

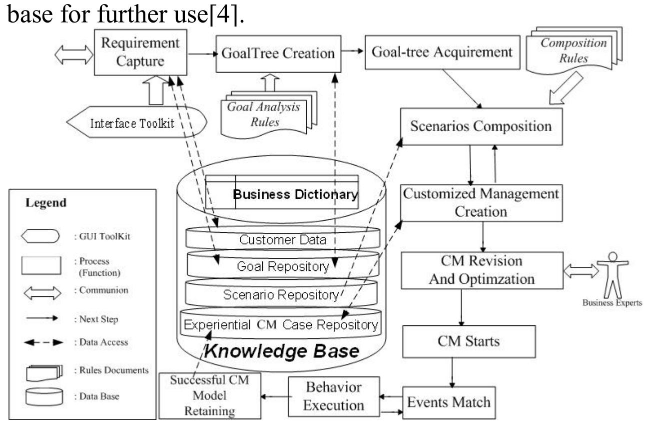

Figure 1. Outline of IPVita works

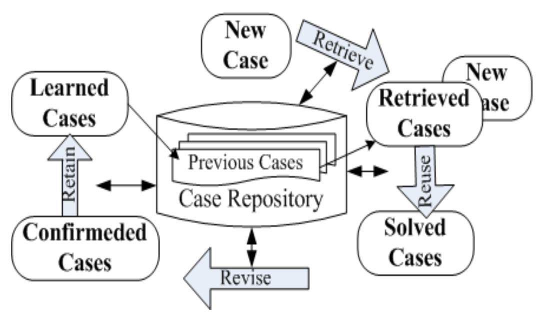

Figure 2. CBR processes cycle

We applied the CBR technique to the travel sector because for travel service companies with many service MC, it will be a good solution to adopt previously effective MC to quickly create new ones and ensure their reliability. There have been some applications of the CBR technique for modeling MC in workflow management system (WFMS) or management system[5,6,7]. Limthanmaphon[8] presented a model of Web Service composition using CBR to run a smart e-business or to provide an efficient Web Service. Marir[9] presented a new approach for managment redesign (BPR) with manually selected goals and targets.

There are some differences between these previous reports and the present study. For example, the features used

978-0-7695-4608-7/12 $26.00 © 2012 IEEE
DOI 10.1109/ISdea.2012.428

for searching similar cases and the related matching algorithm of each work are different. A challenging problem in current CBR research is how to optimize and abridge an ever-increasing experiential case database to improve the efficiency of searching for suitable experiential cases[10].

III. MC MODEL

We suggest that CM model should be organized by a set of independent scenarios, which is the comparatively independent software unit (or manual operation) and comprises a series of behaviors driven by real-time events.

Event: An event is a single point in time when something happens. Events are treated as semaphores, which initiate a state transition.

Definition 1 Event = <E_ID, Name, Time, RelationID, Rank>

Behavior: A behavior is the fine granularity of activity, specifying the basic goals to be achieved, together with a number of roles required, the cost, and the resource and constraint specifications. It is executed by an agent that may be a person or a software program module, and can eliminate inbound events and generate outbound ones. Behavior has the attributes of default, alternative, or optional.

Definition 2 Behavior = <B_ID, Goal, Role, InboundEvent, OutboundEvent, Constraints, Cost, Attribute>

Scenario: A scenario consists of several behaviors. It is used to achieve a specific goal by implementing those behaviors. It also has attributes of default, alternative, or optional. In our study, we propose that scenarios are independent service units and implemented or provided by contracted travel services of virtual corporations according to their specialties.

Definition 3 Scenario = <S_ID, BehaviorsList, Goal, Constraints, Attribute>

Goal: Goal can be achieved by a behavior or a scenario, and it will rely on or consume some resources.

Definition 4 Goal= <G_ID, Name, Resources, Rank, Attribute>

It is a quintuple. Rank is its layer in the goal net, which includes all design goals of a CM model. Attribute has a value of "on" or "off", which means that this goal can be considered or not.

GoalTree: GoalTree describes the requirements of a CM model.

Definition 5 GoalTree = <GT_ID, Name, Constraint, ContributionList, PopularityDegree, Weight>

Every goal node in a GoalTree is a quintuple. Weight denotes the importance of a node. Each node has some characteristics that contribute to the business sub-goals. It has constraints such as time to spend, cost, and so on.

CM model: A CM model is a list of scenarios that have determinate relations.

Definition 6 CM= <C_ID, Name, Scenarios, Relationship, Goal, GoalTree>

Definition 7 Relation = <a, Si, Sj>

a ∈ A={//,→, ⇨ ,⇔, ⊗, ⊙ , ◎ }[11] is symbol of association. Si, Sj ∈ (Scenario Set). Every symbol of association represents a relation between two scenarios.

//: Parallel association

→: Prerequisite association

Si → Sj: The prerequisite relation means that one scenario has to finish before the other starts. Scenario Si has to finish before scenario Sj starts.

⇨: Parallel-prerequisite association

Si⇨Sj: Here, Si presents at the same time as Sj, but Sj has to wait for the result from Si before completing its process.

⇔: Parallel-dependency association

Si ⇔ Sj: Here, Si and Sj progress simultaneously, but the results of each scenario need to be coordinated with the other. This kind of association needs interface-negotiation and deadlock-avoidance mechanisms.

⊗: Overlapping association

Si ⊗Sj: Here, Si has some capacities that are the same as Sj. To compose this overlapping association, the overlapping parts from the scenario that cost more need to be excluded.

⊙： Mutually exclusive association. ◎: Incorporate association.

Thus far, we have defined a CM model. The next section illustrates how such CM is created by the CBR technique.

IV. USING CBR TECHNIQUE FOR CM MODEL CREATION

A. Similarity measure

The similarity measure is a function that evaluates the similarity between a given query and cases in the case base. It measures each attribute or the dimension of the level of difference between a query and existing cases. Most CBR techniques use a generalized, weighted similarity measure such as

$$SIM(x, y)=\frac{\sum_{k=1}^{n} w_k * atr\_sim(C_{ik}, C_{jk})}{\sum_{k=1}^{n} w_k} \quad (1),$$

where Ci and Cj are two cases, wk is the weight or importance assigned to attribute k, and atr_sim(Cik, Cjk) is the degree of similarity between the value of attribute k in cases i and j.

For GoalTree cases, we must first match the tree structure between the query case and the cases in base. If there are isomorphic GoalTrees, then we match their nodes contents farther. As for the ScenarioList case, the similarity measure is based on the following rules except for SIM(x, y):

Rule 1: If the value of the attribute A of the query is exactly the same as the value of the feature F of the case, then the similarity of this attribute is the highest value (equals 1).

Sim(qA, cF)=1, if (qA=cF)

Rule 2: If the attribute A of the query is close to the feature F of the case according to KB, then the similarity is ranked from 0.1 to 1.0 depending on their function similarity in KB.

Sim(qA, cF)=θ, if θ=[0.1, 1.0], SimKB (A, F)= θ

Rule 3: If the attribute A has not been able to match any feature F from the case, then the similarity has the lowest value and equals to 0.

Sim(qA, cF)=0, if (qA≠cF)

The CBR process will start to retrieve the high degree of similarity cases from GoalTree Case because GoalTree is considered the basic feature of CM model. If the degree of similarity is high enough or the GoalTree of the query exactly matches the existing case, then the system will provide the CM model implementation of that selected case as a result. Otherwise, the retrieval process will start again by using the ScenarioList Case instead. ScenarioList Case is a collection of cases which GoalTrees have higher similarities, compared with a liminal value in the first matching stage. It is created on-the-fly. The retrieval process will try to match the query's ScenarioList with that in the ScenarioList Case. Lastly, the retrieval process will suggest a case of CM model, which has the closest GoalTree and ScenarioList similarities to the requested CM. This is the new solution that will be revised by some staff members. It may be retained after being executed successfully.

B. Matching ScenarioList

To retrieve the closest case from the ScenarioList Case, we use rules 1~3 and Formula (1) and aided by the KB. Each designed scenario of query CM model is compared with that of cases from the ScenarioList Case. ScenarioList matching is composed of three processes.

For each case in the ScenarioList Case,

First, extract its scenarios and compare every scenario orderly with the query case but ignore its tn (it is only an order). This matching process can evaluate a similarity degree, SSim.

Then extract its relationship and compare every scenario association pair in it with that of the query case. This matching process can evaluate a similarity degree, RSim.

The compositive similarity degree of ScenarioList is SSim*0.7+RSim*0.3.

C. Retrieval experiential CM

By now, we can implement the retrieval process through the following steps:

Select the initial values of α, β, D according to practical experience. *α* is the lower threshold of content similarity. *β* is the upper threshold of GT(Goal Tree) number to be retrieved from repository. *lev* is the upper height to be matched between GT_0 (Suppose the GoalTree of a query case is GT_0) and GT in repository. *D* is a fixed finite set and $|D| >> lev$ which is used in Algorithm StructureFilter. Algorithm FinalMath invokes Algorithm StructureFilter to get the isomorphic GT set of GT_0, and invokes Algorithm ContentMatch to get the content matched GT set of GT_0 among the result of Algorithm StructureFilter. If the result size of Algorithm ContentMatch is larger than *β*, then let *lev*=*lev*+1 and re-iterate Algorithm FinalMath.

Step 1: Set counter=0; call FinalMath(GT0, GTSet, lev, α, β, D), output matchGTSet.

Step 2: matchGTset includes n（0<=n<=β）GoalTree cases with similarity degrees higher than α, calculated by Formula (2):

$$\mathrm{ContentSim}(GT_0,GT)^{lev} = \frac{\sum \textit{weight of vertex at height lev whose ancestors and itself all matched between } GT_0 \textit{ and GT in sequence}}{\sum \textit{weight of all vertex at height lev in GT}} \quad (2)$$

If n>0 Then the cases in matchGTSet are checked by staff members of the company to determine if there are satisfied results;

If the GoalTree of query exactly matches the existing case or the degree of similarity of GoalTrees is high enough

Then CBR process succeeds; go to Step 6.

Else If counter=0

Then set counter=counter+1, modify β and α, let β=β*(1+30%), α=α*80%, call FinalMath(GT0, GTSet, lev, α, β, D) again and iterate Step 2.

Else CBR process failed; stop.

Step 3: For each GoalTree in matchGTSet, retrieve its scenarios with its relationship from KB to which it belongs, and create a *ScenarioList Case* repository provisionally.

Step 4: Call the ScenarioList Matching process and get the highest similarity cases.

Step 5: A suitable case is selected by the staff members of travel service.

Step 6: Retrieve the CM documents of the selected case, CM_0.

Step 7: Revise and optimize the CM_0, if necessary, in order to create a new "solution".

Figure 3 shows the retrieval process architecture.

V. EXPERIMENTS AND DISCUSSION

By inputting different query case, we tested the method of CM model creation, which uses CBR technology for experiential CM model reuse purposes. In analyzing the experimental results, we demonstrated that if there are exactly the same cases as the query case in the GoalTree Case, then the retrieval process can retrieve them in the first stage, that is, only by matching the GoalTree.

If there are some GoalTrees in the experiential CM repository that are identical to those on the third-level nodes of the query GoalTree, then the probability of finding these GoalTrees is greater than 96% through two retrieval process stages. These nodes are explicit business goals, so the corresponding experiential CM models will be very useful. Quite often, there may be unmatching nodes at the fourth level, in which case it is necessary for the staff members to make a selection or to modify their goal(s) so that the corresponding experiential CM model can be used. Otherwise, they need to change the selected experiential CM model manually.

This experimental system was built on four independent simulative IPVitas, located on four hosts respectively. Each IPVita represented a contracted travel service, that is, a site, and has its own KB constructed in Protégé. We designed hundreds of experimental data to store in every repository in KBs depicted in figure 1. After a KB has been established, its experiential CM case repository was mapped into a local relational database. GoalTree and ScenarioList matching processes run in one of such IPVita systems but based on four KBs. In these processes, transferring data are comparatively small, so it will not be a problem even if the sites are more in practical system.

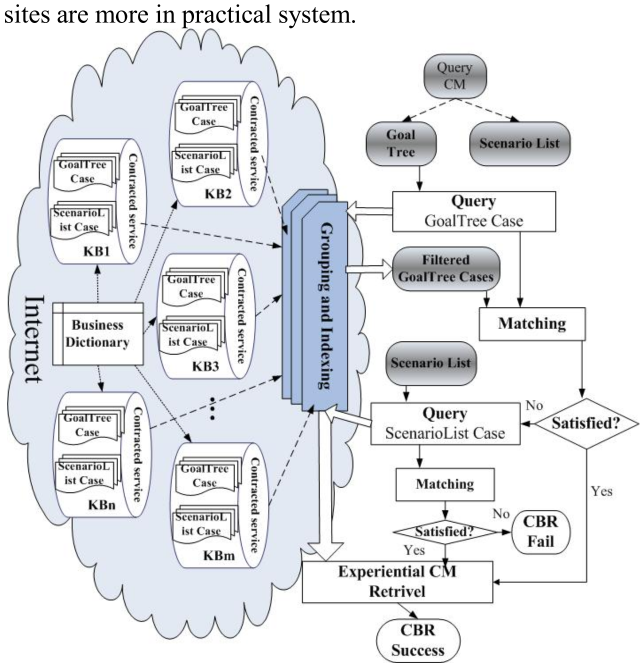

Figure 3. Retrieval process architecture

VI. CONCLUSION AND FUTURE WORK

We set out to develop a CM model capable of facilitating a specific management method in a Web Service environment. The paper first discusses the demand for CM and analyzes its characteristics. Then a CM model with its combined elements is introduced. After describing how to implement CM through the CBR technique, we present our experiment processes and discuss the result.

While believing that we have made progress in exploring a advanced management methodology suitable for a customized mode, we also understand that there is much still to be accomplished. In particular, the CM model still needs to be optimized, and further, there is a need for a better analysis and modeling technique for the design of the CM model query ScenarioList. Work is continuing on both of these aspects.

Acknowledge Part of this work is supported by the National Science Foundation (61173140), the Science Foundation of Shandong Province Project (Y2008G28), and the Independent Innovation Foundation of Shandong University (2010JC010).

REFERENCES

[1] C.H.Crawford, G.P.Bate, L. Cherbakov, K. Holley, C. Tsocanos,Toward an On Demand Service-Oriented Architecture. IBM Systems Journal, Vol.44(2005), P.81

[2] A.Aamodt, E. Plaza, Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications, Vol.7(1994), P.39

[3] R.L.De Mántaras, D.Mcsherry, D.Bridge, et al., Retrieval, Reuse, Revision, and Retention in Case Based Reasoning, Knowledge Engineering Review, Vol.20(2005), P.215

[4] S.Craw, N. Wiratunga, R.C. Rowe, Learning Adaptation Knowledge to Improve Case-Based Reasoning. Artificial Intelligence, VOL.170(2006), P.1175

[5] J.Cardoso and A. Sheth, Adaptation and Workflow Management Systems. (In Proceedings of International Conference WWW/Internet, Portugal 2005)

[6] S. Lajmi, C. Ghedira, K. Ghedira, D. Benslimane, How to Compose Web Services via Case Based Reason. (Conference on e-Business Engineering, China 2006)

[7] T. Osman, D. Thakker, D. Al-Dabass, Bridging the Gap between Workflow and Semantic-Based Web Services Composition. (IEEE/WIC/ACM International Conference on Web Intelligence, France 2005)

[8] Information on
http://crpit.com/confpapers/CRPITV17Limthanmaphon.pdf

[9] F. Marir, S.L. Mansar, An Adapted Framework and Case-based Reasoning for Business Process Redesign. (In Proceedings of 2nd International Conference on Information Technology: Research and Education, London,2004)

[10] H. Li, D.W. Hu, T.Y. Hao, W.Y. Liu, X.P. Chen, Adaptation Rule Learning for Case-Based Reasoning.(In Proceedings of Semantics, Knowledge and Grid, Third International Conference, China, 2007)

[11] Information on
http://crpit.com/confpapers/CRPITV17Limthanmaphon.pdf