Journal of Computational Information Systems 5:1 (2009)
Available at http://www.jofcis.com

A New Business Process Model in Web Service Environment

Qing YAO[1†], Yuqing SUN[1], Haiyang WANG[1]

[1] *College of Computer Science & Technology, Shandong University*
Jinan 250101, China

Abstract

This paper introduces the background and motivation of a new business process model, customized flow model, applied to a web service environment. It then presents a prototype system (Intelligent Platform of Virtual Travel Agency, IPVita) developed for this research, narrates the IPVita's architecture and functions. After giving definitions of the customized flow model, the paper describes how to analyze and capture customer's requirements and how to generate a customized flow according to these requirements.

Keywords: Software Architecture; Process Modeling; Customized Flow; Web Service; Goal Oriented

1. Introduction

The research discussed in this paper focuses on Process Factors[1,2,3] of BPM(Business Process Management) software in a web environment, and explores new process application model based on Web Services. In this paper, business process is named as flow. We have developed an Intelligent Platform of Virtual Travel Agency, IPVita, as a prototype system. Under this platform, flows are executed as customized flows. Main functionalities of the platform are: help registered customers with travel destinations and service requirements, and generate travel route accordingly, and eventually create customized flow of service. This flow then runs throughout the customer's actual trip. The flow is a customized one since it is generated for one customer or a group of customers with exactly the same itinerary, and it is executed in a Web Service environment.

The structure of this paper is as follows. Section 2 gives the architecture of IPVita, and then Section 3 describes the customized flow model. In Section 4, we depict how to capture customer's requirements and generate service flows. Finally, Section 5 provides experiments of some process, and Section 6 concludes the paper and indicates key topics for future research.

2. Architecture

The IPVita platform architecture can be illustrated by Figure 1[4]. The platform consists of four components: 1) User Interface. 2) Flow Generation. 3) Web Service Binding for Flow. 4) Repository Management.

- **User Interface:** it is used for customer requirements gathering (i.e. travel goal gathering) and maintenance of the Domain Repository by the domain experts in travel services. The main functionality of the customer requirements gathering interface is to help the customer to identify the goals based on the Goal Repository. There are two possibilities here. a) The customer is very clear about his/her goals and

[†] Corresponding author.
Email address: yaoqing@sdu.edu.cn (Qing YAO)

requirements, is able to interact with the platform and follows the step-by-step suggestions to get a goal-tree. b) The customer can only provide some restrictive conditions or personal preferences. In this case, the platform needs to provide multiple iterations of heuristic communion by utilizing existing goal knowledge to ultimately achieve a satisfactory goal-tree. The final product is a goal-tree representing the customer requirements. This goal-tree is the foundation for generating a customized flow.

On the other hand, the domain expert interface allows these professionals to maintain the Domain Repository, which includes Scenario Repository, Goal Repository and Experiential Flow Repository.

- **Flow Generation:** Once the goal-tree is available for a customer, a service flow is generated and modified by searching the Scenario Repository. This software component consists of a searching and assembling algorithm and an interface management program for customers and domain experts.
- **Web Service Binding of Flow:** We have developed a web service search and match binding algorithm based on the existing web service composing technique. At present the algorithm is a very simple one; it is only used for functional testing of the platform and will be refined and improved later.
- **Repository Management:** This involves the enrichment and optimization of the Scenario Repository and the Goal Repository, and the maintenance and optimization of the Experiential Flow Repository. The already-developed Customer Goal Analysis Algorithm, although very simple, has some self-learning capabilities. Generic Algorithm will be adopted to enhance the self-learning and optimization features of the Goal Repository.

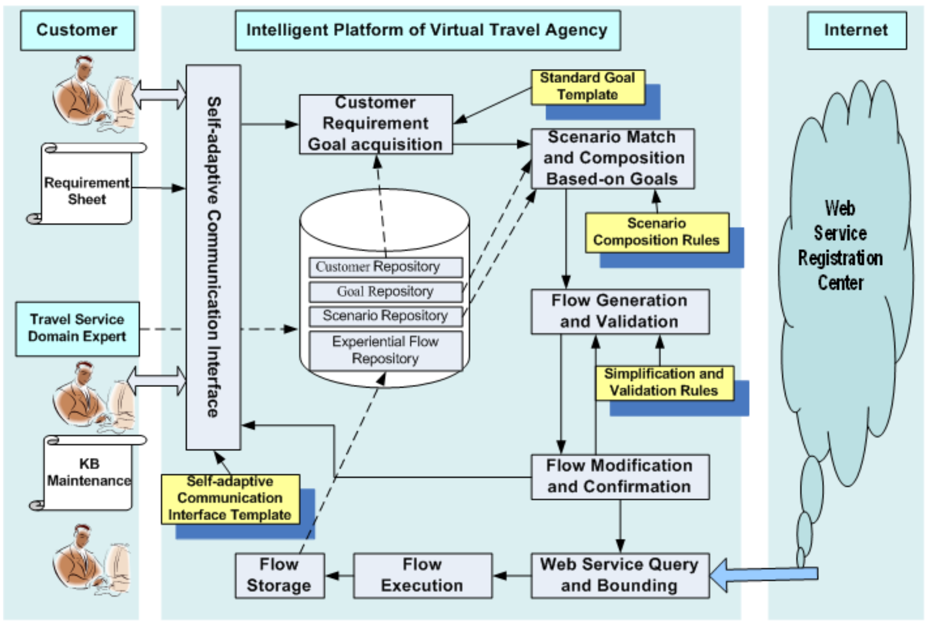

Customer
Intelligent Platform of Virtual Travel Agency
Internet
Standard Goal Template
Customer Requirement Goal acquisition
Scenario Match and Composition Based-on Goals
Requirement Sheet
Self-adaptive Communication Interface
Scenario Composition Rules
Customer Repository
Goal Repository
Scenario Repository
Experiential Flow Repository
Web Service Registration Center
Flow Generation and Validation
Travel Service Domain Expert
Simplification and Validation Rules
KB Maintenance
Self-adaptive Communication Interface Template
Flow Modification and Confirmation
Flow Storage
Flow Execution
Web Service Query and Bounding

Figure 1 Outline of the Intelligent Platform of Virtual Travel Agency

3. Customized Flow Model

We begin with the definition of the flow model, because of its crucial effect on the entire software framework. We suggest that flows should be organized by a set of relatively independent scenarios, comprised of a series of behaviors driven by real-time events to perform their independent functions.

Event: An event is a single point in time when something happens, according to Allen's conception of temporal semantics[5]. Events are treated as semaphores, which initiate a state transition.

Definition 1 Event = <E_TypeID, Name, Time, RelationID, Rank>.

Event is a 5-tuple. E_TypeID is a special type identifier of an event, denoting the domain to which an event belongs. Name is the title of an event, such as payment, no air ticket, or login. Time refers to the beginning time of an event. RelationID indicates the event relationship space that depicts all dependent relationships of events involved. Event is assigned a rank (weight) in a flow to represent its importance in the flow.

Behavior: A behavior is a fine granularity of activity, specifying the basic goals to be achieved, together with a number of roles required, the cost, and a number of resource and constraint specifications. It is executed by an agent that may be a person or a software program, and can eliminate inbound events and generate outbound ones. Behavior has attributes of Default, Alternative, or Optional. See Definition 4 for Goal. Examples include book air ticket, check suitable hotels and local entertainment. It is a 8-tuple.

Definition 2 Behavior = <B_TypeID, Goal, Role, InboundEvent, OutboundEvent, Constraints, Cost, Attribute>[6].

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 software framework, scenarios are primarily implemented by web services.

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

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

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

It is a 5-tuple. Rank is its layer in the goal net, which includes all design goals of a customized flow. Attribute has a value of "on" or "off", means that this goal can be considered or not.

GoalTree: GoalTree describes the requirements of a customer. Every goal node is a 5-tuple. Weight denotes importance of a goal. Each node has some characteristics that contribute to customer preferences in his(her) travel, such as shopping, sport, religion, time consumption, and cost etc.

Definition 5 GoalTree = <G_TypeID, Name, Constraint, ContributionList, Weight>

Root node is an integrative goal, we call it maintenance goal, which reflects the customer's trip-specific interests, requirements and restrictive conditions (e.g. cost, choice of transportation means, etc.). The names of the cities or regions along the travel routes are sequentially stored in the second-level nodes, and we call it achievement goal. An achievement goal includes maintenance sub-goals; conversely, a maintenance sub-goal can place a constraint on an achievement goal. There are many achievement goal nodes at the third level, where travel items within a given city (region) are stored.

Flow: A flow is a list of scenarios that have determinate relations. It is a 6-tuple.

Definition 6 Flow = <F_TypeID, Name, ScenarioList, RelationList, Goal, GoalTree>

F_TypeID is a string symbol that describes what business type the flow belongs to. RelationList is a list composed of relations, as in definition 7.

Definition 7 Relation = <a,Si,Sj>

a∈A={→,⇨,⬄,⊗， ⊙， ◎}[7], is symbol of association. Si, Sj∈(Scenario Set). Every symbol of association represents a relationship between two scenarios, e.g., ((→,Paying_000,Booking_010), (⇨,Reserving_020,Flighting_000), (⊗,Car_renting, Hotel_service_000)……). Paying is a prerequisite for booking. Reserving should be executed in advance or in parallel with arranging flights. Car_renting can overlap with Hotel_service.

→: Prerequisite Association

Si → Sj: The prerequisite relationship 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 in parallel (simultaneously), but the results of each scenario need to be coordinated with the other.

⊗: 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 flow model, which is the key part of the software framework. The next section depicts how to generate such a flow.

4. Customer's Requirements Capture and Service Flow Generation

4.1. Goal-Analysis Algorithm

Input: maintenance goals and restrictive achievement goals.

Output: customer's goal-tree, GT.

Step1: Create a null customer goal-tree GT0.

Step2: Communicate with the customer: get his/her maintenance goals and restrictive achievement goals.

Step3: Add customer's restrictive achievement goals to GT0

Step4: Search for achievement goals in the Goal Repository, according to the customer's maintenance goals.

Step5: If there exist suited achievement goals,

Then call a Filter Genetic Algorithm to select most suitable goals and add them to GT0.

Else invoke the Self-learning Algorithm and update the Goal Repository, and mark the unfit maintenance elements of the customer's maintenance goals.

Step6: Check if the customer's maintenance goals are met. If it is OK, go to step 8.

Step7: Communicate with the customer and disassemble his/her maintenance goals, so as to rebuild them. Go to Step 4.

Step8: Adjust GT0 by aggregating contributions from all leaf nodes of GT0, and then compare the result with the customer's maintenance goals. Delete those leaf nodes that have lower contribution values until the result matches the maintenance goals.

Step9: Determine the validity of the Goal-Tree. Follow the goal analysis rules to check GT0 for location continuity and time continuity. If the result is invalid, go to Step 7; otherwise, let GT= GT0.

Step10: Output GT.

4.2. Method of customized flow modeling

4.2.1. Extended EPCs

EPC are an intuitive graphical business process description language introduced by G.Keller[8]. It is the modeling tool used by the Architecture of Integrated Information System(ARIS)[9]. EPC has a strong ability of model expression and is easy to understood, so it is widely used in BPR and workflow definition. At the present time, for meeting the requirement of different models, EPC is extended to several modes[10]. In this paper, based on the requirement of customized flow, EPC is extended by adding two elements: goal and variable attribute. We call it Customized Extended-EPC(CE-EPC).

Definition 8. CE-EPC=<E，P，C，T，A，G，a>

It is a seven-tuple. In which, E is a finite set of events. P is a finite set of functions. C is a finite set of logical connectors. $T \in C \rightarrow \{AND, OR, XOR\}$ is a function which maps each connector onto a connector type. $A \subseteq (E \times F) \cup (F \times E) \cup (E \times C) \cup (C \times E) \cup (F \times C) \cup (C \times F) \cup (C \times C)$ is a set of arcs. G is a set of goals. "a" is a value describing the variable attribute of P (on or off).

4.2.2. Process of the Customized Flow Modeling

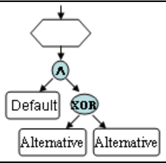
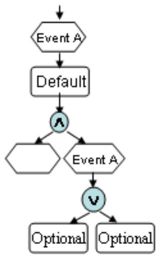
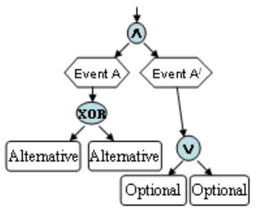
Goal-Analysis Algorithm catches the fundamental goals of customers through interaction with him(he), that is, GoalTree[11]. Then this GoalTree is transformed to design goals (see definition 4) according to design goal template, which are divided into independent small granularity goals in order to reduce the complexity.

The steps of generating customized flow can be summarized as follows: First, for a certain independent goal in the design goals, search in the scenario repository to get the useful scenarios and the relationships between them (shown in the definition 7). Hereby, we can get all the scenarios of each goal. Partial CE-EPC of each scenario is created through the Modeling-SubFlow process. Then, integrate the partial CE-EPCs of every goal through the Merging-SubFlow process to create a flow.

(1) **Modeling-SubFlow**: a single scenario is mapped to CE-EPC by analyzing and linking behaviors in it. This algorithm is guided by the following rules.

- Behaviors are linked one by one base on the time order and mapped to the function P of CE-EPC.
- The inbound events and outbound events of behaviors are mapped to the event E of CE-EPC.
- The single event is regarded as single input of P.
- For the behavior of more than one input or output event, the events are jointed by the connector XOR as one input or output E.
- For the variable attribute of behavior triggered by a certain event, according to the attribute value, the mapping rules are shown in table 1.

Table 1 the mapping rules from the variable function triggered by a certain event to EC-EPC

Function attribute	**Mapping rules**	**Example**
All default	All of the behaviors are jointed by the connector Cs-And (Cs: one father node，many son nodes)	Omit
All Alternative	All of the behaviors are jointed by the connector Cs-XOR	Omit
All Optional	All of the behaviors are jointed by the connector Cs-OR	Omit
Default and Alternative	All of the default behaviors and a XOR connector which joint all the alternative behaviors are jointed by the connector And.	
Default and Optional	Firstly, the default behaviors are set, and the input event triggering the optional behavior is added to the output event of the default behavior. Then add an OR connector following that event to joint all optional behaviors.	
Alternative and Optional	Firstly, copy this event and give it another name (they are two different events now). Then joint them by a Cj-And connector. Finally, the two events are followed respectively by the alternative behaviors and optional behaviors. （Cj: many father nodes，one son node）	 Event A Event A' XOR Alternative Alternative Optional Optional

After the initial CE-EPC is constructed, business experts examine the logical rationality based on the completion of goals from the business point of view. At the same time, the variable attribute of P is assigned a value.

（2）**Merging-SubFlow**: scenarios(SubFlow) are merged and integrated.

Every selected scenario will be transformed to a CE-EPC(subflow). The following work merges the CE-EPC of scenarios. We designed an algorithm and some rules to merge and reconstruct the scenarios.

Following is its main steps:

According to the relationship of different scenarios, the merging method is different.

- For Prerequisite Association <→,Si,Sj>: the output event of Si and input event of Sj are simply jointed by Cs-And, then linked with Sj.
- Parallel-Prerequisite Association<⇨,Si,Sj>: means that there are relationships between the output events of Si and some events in Sj. In other words, the output events of Si are the input events of some behaviors in Sj. Refering to the event relationship space, search out the dependent events of Si’s output events. These events in Sj are the joint point of the two scenarios.
- Parallel-Dependency Association <⇔,Si,Sj>: the CE-EPC of the two scenarios need to be integrated and reconstructed completely. We adopts the algorithm of merging two EPCs introduced in reference[12].
- Overlapping Association <⊗,Si,Sj>: the overlapping parts of the scenarios that cost more need to be exclude. So the overlapping part of a scenario is deleted, then they are jointed through the same way as Prerequisite Association.

Through the above two processes: Modeling-SubFlow and Merging-SubFlow, a customized service flow is generated initially. Next work is to simplify and verify the flow. We adopt the thought of transforming EPC to Petri Net[13, 14] and then verify the Petri Net’s rationality. The verifying rationality algorithm checks the flow’s state space through automata theory method.

5. Experiments

We have manually compiled about 350 business goals for the travel domain, and created the Goal Repository according to the hierarchical relationships among the goals. The Scenario Repository, which stores about 100 scenarios with their embedded behaviors, is created the same way. The Experiential Flow Repository is partially populated in a similar fashion with about 150 virtual experiential flows.

At present, experiments focus on two issues: 1) Customer goal analysis(Algorithm Goal-Analysis) and 2) Modeling-SubFlow and Merging-SubFlow.

5.1. Goal-analysis Algorithm

Currently, this algorithm is not perfect. It is executed while IPVita interacts with a customer, and can run repeatedly until a qualified result is obtained. We found that there are two factors that may determine its success: 1) whether the content of the Goal Repository is sufficiently complete; and 2) whether a customer provides restrictive achievement goals. Satisfying these two factors can assist the Goal-Analysis Algorithm to quickly get a goal-tree that meets the requirements. If, however, the customer leaves out restrictive achievement goals and provides maintenance goals only, then the analysis process becomes rather complex and may require many iterations and the GoalTree generated from each iteration may be different.

5.2. Flow Generation

We designed more than thirty representative GoalTree data to execute Modeling-SubFlow and Merging-SubFlow. Table 2 gives some typical experiment results. It is recognized that with the goals number increasing, business experts’ intervention works are increased as well.

Table 2 Algorithm Modeling-SubFlow and Merging-SubFlow Results

Number of Goal Nodes	**Number of Initial Selected Scenarios**	**Effectual Scenarios**	**Manual Work Ratio**	**Rational**	**Successful After Modification**
8	18	11	10%	YES	YES
6	15	10	10%	NO	YES
10	18	11	35%	YES	YES
12	22	13	20%	YES	YES
14	22	18	20%	NO	NO

17	23	18	20%	YES	YES
20	29	20	30%	YES	YES
24	34	27	30%	NO	YES
25	45	30	30%	YES	YES
28	45	30	32%	NO	YES
32	50	35	40%	NO	YES
34	54	40	40%	YES	YES

6. Conclusions and Future Work

We set out to develop a business process model and supporting software framework capable of facilitating a specific management method for business processes in a web service environment: that is, customized flow. This paper first defines a conceptual model of customized flow with its combined elements is introduced. After describing how to generate a customized flow by capturing customer requirements, we present our experiment processes and discuss the result.

While believing that we have made progress in exploring BPM methodology suitable for a web services environment, we also understand that there is much still to be accomplished. In particular, the flow model still needs to be optimized, and further, there is a need for a better requirement analysis method that addresses challenges unique to customized business process management applications. Work is continuing on both of these aspects.

Acknowledgement

This work is supported by NSFC (National Natural Science Foundation of China) under Grant No. 60673130.

References

[1] F. Q. Yang, H. Mei, J. Lu, Z. Jin, Some Discussion on the Development of Software Technology, Acta Electronica Sinica, 2002, 30(12A): 1901-1906.

[2] Z. L. Chen, Q. X. Wang, H. Mei, F. Q. Yang, Dealing with the variability in object-oriented domain design. Acta Electronica Sinica, 2001, 29(11): 1486-1490

[3] S. H. Liu, J. Wei, T. Huang, A Dynamic Process Model Based on service Cooperation Middleware. Journal of Software, 2004, 15(10): 1431-1440.

[4] Q. Yao, L. Z. Cui, H. Y. Wang, Toward Cooperative Designing of Customized Business Process in Web Service Environment, Computer Supported Cooperative Work in Design, 2007

[5] Sutcliffe, A. (1998) Scenario-Based Requirement Analysis. Requirements Engineering, **3**, 48 - 65

[6] Henricksen, K. and Indulska, J. (2004) A Software Engineering Framework for Context-Aware Pervasive Computing. Proceedings of PERCOM'04, Orlando, Florida, Mar. 14-17, pp. 77-86, IEEE Computer Society, California

[7] Limthanmaphon, and Y. Zhang, B. (2003) Web Service Composition with Case-Based Reasoning. The Fourteenth Australian Database Conference (ADC2003), Adelaide, Australia, Inc.Vol.17:201-208, Australian Computer Society, Australian

[8] Van Der Aalsta, W.M.P.（1999） Formalization and verification of event-driven process chains. Information and Software Technology, 41, 639-650

[9] Scheer, A. W. and Nuutgens, M. (2000) ARIS architecture and reference models for business process management. LNCS/Business Process Management, 1806, pp.301-304, Berlin, Germany

[10] Mendling J. Neumann G. and Nüttgens M.(2005) Yet Another Event-Driven Process Chain, LNCS/Business Process Management, 3649, 428-433

[11] Regev, G. and Wegmann, A. (2005) Where do goals come from: The underlying principles of goal-oriented requirements engineering. 13th IEEE International Conference on Requirements Engineering (RE'05), La Sorbonne, France, Aug.29, pp.353- 362. IEEE Computer Society, Los Alamitos, California

[12] Mendling, J. and Simon, C. (2006) Business Process Design by View Integration. LNCS/Business Process Management Workshops, 4103, pp. 55-64, Berlin, Germany

[13] Mendling J. Neumann G. and Nüttgens M. (2005) Towards Workflow Pattern Support of Event-Driven Process Chains (EPC). 2nd Workshop XML4BPM 2005, Karlsruhe, Mar.1, pp.23-38. Karlsruhe, Germany

[14] Van Der Aalst W. M. P. (2003) Challenges in Business. Process Management: Verification of Business Processed Using Petri Nets. Bulletin of the EATCS, 80, 174-199