2012 International Conference on Information and Computer Applications (ICICA 2012)
IPCSIT vol. 24 (2012) © (2012) IACSIT Press, Singapore

REWARD BASED PROCESS - SERVICE COMPOSITION MODEL USING STATE LINK MATRIX

Baskaran Alagappan[1a], Dr.T.Ravi[2], Danish Kumar[3] and Nithyanandam[4]

[1]Department of computer science, KCG College of Technology, Anna University, India
[2]Head of the Department, Department of computer science, KCG College of Technology
Anna University, India
[3]Department of computer science, KCG College of Technology
Anna University, India
[4]Dept.of Aeronautics, Hindustan University, India
[a] baskarkamal@gmail.com

Abstract. The objective of the work is to propose a formal model for the process and service composition and verify the model based on the rewards that the model can provide to the end user. The work focuses on the user specified cost and completion time of the customized composition representing various business processes and the sub-processes along with their corresponding services. The model considers the "rewards" for each and every state of the services based on the business process requirements. A State Link Matrix (SLM) is proposed to decide the correct choice of composition not only in the process level but also in the service level. The SLM is derived from the coarse grained services in that process and augmented with the next immediate process to satisfy the requirements. The model is formally verified as Continuous Time Markov Chain model using PRISM tool to bring out the minimum and maximum rewards in the composition. The time bound process composition and the cost bound service composition is analyzed with the help of CTMC and DTMC models to explore the probabilistic and stochastic nature of the process– service linkage. The reward based process and service composition model is tuned as per the SLM so as to satisfy the client requirements.

Keywords: State Link Matrix, Business Processes, Stochastic Process Modeling, PRISM, Cost and Completion time, Process Algebra

1. Introduction

In any business process with a support of web services, the set of available services is stable and service requests are uniform, so that static environments as well as static process structures address production workflows. Service compositions can be modeled manually to comply with service requests. In highly dynamic systems, service requests and available services might vary over time. As a result, manually modeled composed services can hardly incorporate a wide range of service requests and changes in the available services, so that an approach based on planning of service compositions is more promising. Web Services are captured by semantically rich descriptions, synthesis and orchestration become more compelling, given a set of component Web Services and a user specification, the synthesis deals with coordinating the component Web Services in order to obtain the specification of a process that exactly matches or that is "as close as possible" to the user specification[1]. Several specifications have been proposed recently, most notably Business Process Execution Language for Web Services [BPEL] for service composition, Web Service Coordination (WS-Coordination) and Web Service Transaction (WS-Transaction) for service co-ordination and transaction respectively [2]. The process model defined by BPEL is based on the WSDL based service description model. The services that the processes invoke/reply using basic activities are represented using their WSDL description.

2. Business Processes

A Business Process (BP) may be considered as an atomic activity or a collection of sub processes which in turn needs some web based services to complete the submitted task.

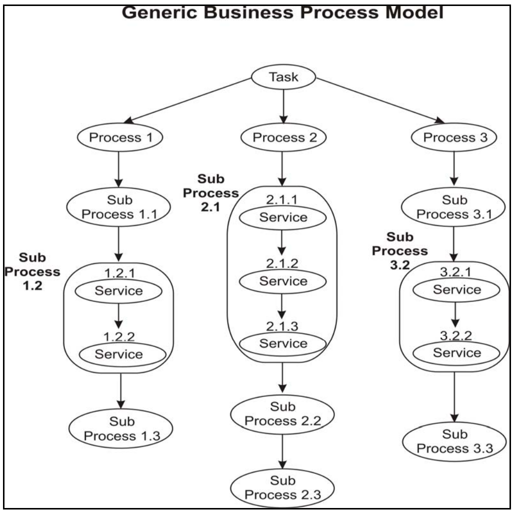


Fig.1.Business Process Model

Service composition allows composing applications and processes using services from heterogeneous software components in different environments without regard to the details and differences of those environments. Service composition is much more achievable if the services are constructed with coarse-grained interfaces. With the help of effectively designed and composed coarse-grained services, a hierarchical business processes and their sub processes can be productively composed for new business applications. The generic business process model that explores the linkage between the sub processes and the associated services in the form of a tree structure is shown in Figure 1. In each of the process synthesis and the involved service composition are to be done as per the web user expectations.

2.1 STATE LINK MATRIX

The services composablity and the rate at which the composition takes place play a very important role in determining the process synthesis. To analyze the behavior of the service oriented computation, a State Link Matrix is proposed where the processes are represented as a matrix in which the transition rates between the states of the associated services are elements.

$$P1=\begin{array}{c} S_1 \\ S_2 \\ S_3 \end{array}\begin{pmatrix} \lambda_{1i} & \lambda_{1d} & \mu_{1r} & \lambda_{1c} & \mu_{1s} \\ \lambda_{1i} & \lambda_{1d} & \mu_{1r} & \lambda_{1c} & \mu_{1s} \\ \lambda_{1i} & \lambda_{1d} & \mu_{1r} & \lambda_{1c} & \mu_{1s} \end{pmatrix}$$

Similarly, the other matrix P2 also denotes the same set of sequencing services as specified by the end user. Many such process can be specified according to the user demands. Given below is an example of the process P1 is shown below wherein which values are substituted for the given rates. These values are choose at random to avoid any standard results. These randomized values are given by the end user.

Another example is being specified which illustrates the process P2. Again values specified by the user are taken into consideration.

G=1/45:(P1 | | P2) (2)

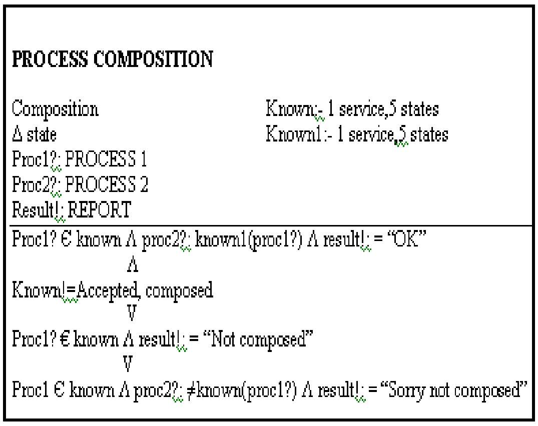
The goal matrix parallel functions the two processes P1 and P2 in the gamma rate of probability.

3. Formal Specification

The formal specification has an action by one process enabled by a predicate on the state of the process, without describing how the first process learns about the second process's state. The process-service composition is formally specified in Z specification language, shown in the Figure 2 and 3 respectively .

The variable parameters are:

- **State:** For each state, the value of the status will be the string denoting whether the node is changeable or not changeable.
- The problem explained here is based on which of the nodes are fully changeable and vice versa. In order to specify the same, state variable is introduced.
- **Proc1:** For each process Proc1, the value of the **Proc1** denotes the command which could either be a request from or a response to another node. Its value is set to"?", if the node in consideration has not yet chosen a command.
- **Known:** For each process, the value of the **known (proc1)** denotes a set of one service and five states which belongs to the set known database.



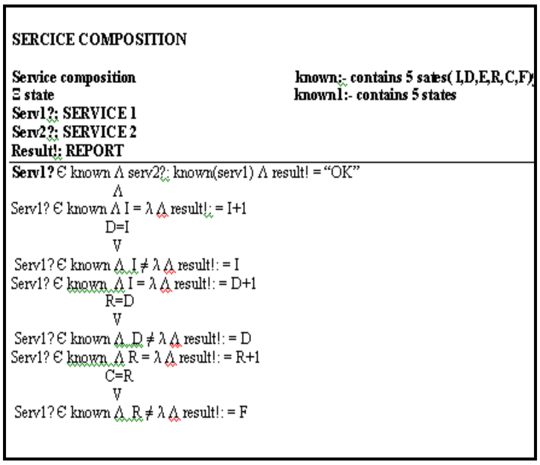
```
PROCESS COMPOSITION

Composition                    Known:- 1 service,5 states
Δ state                        Known1 :- 1 service,5 states
Proc1?: PROCESS 1
Proc2?: PROCESS 2
Result!: REPORT
---------------------------------------------------------
Proc1? Є known Λ proc2?: known1(proc1?) Λ result!: = "OK"
                 Λ
Known!=Accepted, composed
                 V
Proc1? € known Λ result!: = "Not composed"
                 V
Proc1 Є known Λ proc2?: ≠known(proc1?) Λ result!: = "Sorry not composed"
```

Fig.2. Z-Specification of Process Composition

The assertions are:

1. Proc1 (the set of all process) is a finite set that contains one service and five different states.
2. "?" is a command that is not known.
3. Known is a database which contains the service and states.
4. Δ is a positive real number.

The initial condition is that the process1 that is (proc1) which belongs to the database known where the service and states are stored. And at the same time if process 2 that is (proc2) which belongs to the database known1 is satisfied then the two processes are composed. If process1 do not belong to the database known the result is executed as the process is not composed. And there is one more condition if the process1 belongs to the database known but at the same time whereas procees2 do not belong to the database known1 the result is again executed as the process is not composed.



```
SERCICE COMPOSITION

Service composition            known:- contains 5 sates( I,D,E,R,C,F)
Ξ state                        known1:- contains 5 states
Serv1?: SERVICE 1
Serv2?: SERVICE 2
Result!: REPORT
---------------------------------------------------------
Serv1? Є known Λ serv2?: known(serv1) Λ result! = "OK"
                 Λ
Serv1? Є known Λ I = λ Λ result!: = I+1
               D=I
                 V
 Serv1? Є known Λ I ≠ λ Λ result!: = I
Serv1? Є known Λ I = λ Λ result!: = D+1
               R=D
                 V
 Serv1? Є known Λ D ≠ λ Λ result!: = D
Serv1? Є known Λ R = λ Λ result!: = R+1
               C=R
                 V
 Serv1? Є known Λ R ≠ λ Λ result!: = F
```

Fig 3. Z-Specification of Service Composition

The variable parameters are:

- **State:** For each state, the value of the status will be the string denoting whether or the node is changeable or not changeable.
- The problem explained here is based on which of the nodes are fully changeable and vice versa. In order to specify the same, state variable is introduced.
- **Serv1:** For each process Proc1, the value of the **Serv1** denotes the command which could either be a request from or a response to another node. Its value is set to”?”, if the node in consideration has not yet chosen a command.
- **Known:** For each process, the value of the **known (proc2)** denotes five states which belong to the set known database.

The assertions are:

1. Serv1 (the set of all services) is a finite set that contains one service and five different states.
2. “?” is a command that is not known.
3. Known is a database which contains the five states.
4. Ξ is a positive real number.

4. Reward Properties

Once the rewards are associated, properties are specified to tune the reward according to the users requirements. Properties are specified using logical operators in the property window of the Prism tool.

As per the syntax of the model checker it can be specified as follows,

R=? [C<=5.5] (3)

R=? [C<=3.5] (4)

The equation 3 implies the amount of reward associated to a particular service when the cumulative reward is less than or equal to 5.5.The equation 4 implies the same when the cumulative reward is less than or equal to 3.5.A graph may be plotted using this value specified by the user. The user may decide upon the path length, number of transitions, maximum and the minimum time. The expected reward and the maximum time may be plotted in the y-axis and x-axis.

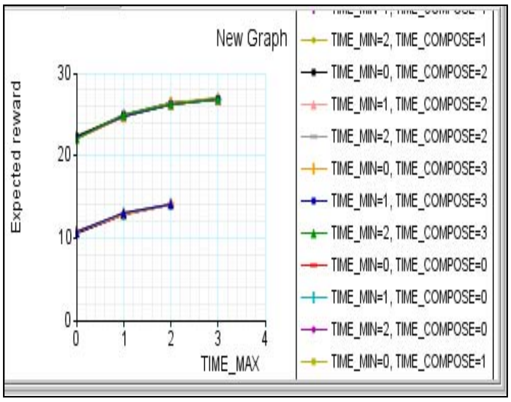


Fig.4. User Requirement-Rewards of 3 Processes

The properties for a particular service composition can be verified and simulated using the tool. As shown in the figure 5, by providing the values for the constant identifiers TIME_MIN,TIME_MAX and TIME_COMPOSE it is possible to study the behavior the reward based model. The above figure represents the graph generated using three processes and different cumulative reward properties. The curve denotes the maximum time taken and the expected number of rewards associated with that service or subprocess.The curves are plotted using the user specified reward properties as in equation 3 and 4.

5. Conclusion

From the above analysis and verification of the proposed model, it may be said that business process composition is parallel and the service composition is asynchronous. Service composition is based on the various linkage elements like lambda, mu and gamma of the State Link Matrix. It is also noted that the number of paths and the corresponding rewards indicate the cost and the completion time of the specified model which in turn is based upon the user expectation. The minimum cost, minimum and maximum completion time may be modified from the client side based on their requirements. Failure in the completion of a service indicates the failed state of either the sub-process or the process itself, minimum rewards results in the best and quick composition, while maximum rewards results in maximum cost for the composition. It is also observed that when the number of service increases the completion time is found to increase. The properties based on the specification of any attribute and accumulation of those attributes may be declared to get the constrained behavior of the model. The complex coarse grained business applications can be derived by suitably changing the number of services and sub processes.

6. References

[1] Automatic Composition of Process-based Web Services: a Challenge, *WWW2005*, May 10–14, 2005, Chiba, Japan.

[2] Business Process Execution Language for Web Services, version 1.1, May 2003, www.ibm.com/developerworks/library/ws-bpel/

[3] Model Checking Continuous Time Markov Chains Adnan Aziz The University of Texas at Austin And Kumud Sanwal Lucent Technologies And Vigyan Singhal Tempus-Fugit, Inc.and Robert Brayton The University of California at Berkeley

[4] M. Koubarakis, D. Plexousakis; ”A Formal Framework for Business Process Modeling and Design”, Information Systems, Vol. 27, pp. 299-319, 2002

[5] Service-Oriented Computing,Martin Bichler,Technische Universität München Kwei-Jay Lin,University of California, Irvine,"Composition and QoS issues present new challenges in business services networks".IT System Perspective,March 2006,pp.88-90

[6] "Eliciting Service Composition"in a Goal Driven Mannerby Rim Samia Kaabi,Carine Souveyet and Colette Rolland.

[7] www.ibm.com/ebooks

[8] www.research.ibm.com/journal/sj/444