

A Goal-Directed Orchestration Approach for Agile Service Composition

V. Portchelvi

Dept. of Banking Technology, School of Management, Pondicherry University, India
Email:portchelvi_vasu@yahoo.co.in

V. Prasanna Venkatesan

Dept. of Banking Technology, School of Management, Pondicherry University, India
Email:prasanna_v@yahoo.com

Abstract— Composition of services provides value added service by combining existing services and is essential to meet the varying users' requests. The need for on-demand, automated, on-the fly and failure resilient service composition led to various dynamic and adaptive service composition approaches. An overview of several existing composition approaches is provided and the limitations in these approaches are identified and depicted as research opportunities. It has been found that all these approaches behave in a rigid way to respond to the changing services environment. They are bridged by proposing a Goal-Directed Orchestration approach which employs an orchestration engine to provide flexibility in responding to the changes in dynamic services environment. To illustrate how our approach could work better than the other existing approaches, we discussed with a usage scenario in travel trip planning domain. Our proposed model is compared with the existing models based on a set of defined features.

Index Terms — Dynamic Service Composition, Composition Goal, SOA, AI Planning, Goal Model, Orchestration Engine

I. INTRODUCTION

Organizations build complex services by adopting service composition strategy in SOA. Service composition supports B2B and EAI by combining the services inside and outside the organization [1]. Service composition is a process in which multiple services are identified, arranged, bound and executed. The composition process has multiple steps and if all the steps in the process are automated then it is called fully-automated service composition. The need for composition arises when a request to a service is not satisfied by a single service then multiple services are identified and composed automatically (composite service) to satisfy the request. Different requests may arise which leads to generation of customized composite services. To satisfy varying requests, services are to be composed dynamically for each request. This needs an on-demand automated dynamic service composition framework. This would benefit a number of application areas such as e-commerce and systems integration. To illustrate, in e-commerce, an automated planning system capable of discovering and interacting with flight itinerary and accommodation booking service could automatically arrange a travel trip based on the user request [1]. Plenty

of Dynamic service composition approaches are available in the literature and are classified into two major categories namely AI Planning based approaches and Goal-Driven approaches [2].

The first part of the paper discusses about the composition process, challenges, solutions and research gap in dynamic service composition. The subsequent part details about the proposed problem of agile service composition and Goal-Directed Orchestration (GDO) approach as a solution.

II. Dynamic Service Composition Approaches

A new value added service (composite service) can be created by aggregating the existing services and the aggregating process is called service composition. Such composition could be realized through web services technology. Though web services environment provide a way to realize composition, they are characterized as highly dynamic and pose several challenges to the composition process [3]. In order to tackle the dynamic services environment, the composition process must be flexible enough to modify, extend and adapt changes at runtime and this requirement has led to the dynamic service composition process. There exist two major approaches that achieve dynamic service composition namely AI-Planning based approaches [4] and Goal-driven approaches [5]. In AI Planning based approaches, Dynamic web services composition problem is mapped to an AI Planning problem. Planning refers to the task of finding a sequence of actions that will achieve a given goal. An AI Planning problem is represented as states, actions and goals. A request to the planner in these approaches is specified in terms of initial state, goal state and constraints. In AI-Planning based dynamic web services composition approaches, web services are modeled as actions, a plan generated combining these actions satisfying a request represents a composite service and the desired service as a goal. A request to the composite service contains the specification of initial state, goal state and possibly optimization criteria. Generally, a composition request is expressed in terms of Inputs, Outputs, Preconditions and Effects (IOPE) [6]. The composite service is built based on the request that specifies the required service automatically.

In Goal-Driven approaches construction of customized composite service based on the request is reflected by the goal concept. The main element in these approaches is the concept of goal. A goal provides the necessary information to the intelligent mechanism for automated problem solving. A composition goal tries to capture the user’s needs and analyze them to derive the requirements for the composition to be modeled. It carries all the required information for automated discovery, composition and execution of services [5]. For this to be made possible, the composition goal and services are to be described in semantic and machine understandable format. Then the composition goal is transformed into an appropriate combination of component web services.

III. DYNAMIC SERVICE COMPOSITION PROCESS

Current research has given a view that the dynamic composition process has various stages in developing a composite service and has led to various approaches. This section discusses about the stages of composition process in the above mentioned two predominant approaches namely AI-planning based and Goal-driven approaches in detail.

A. AI-Planning Based Composition Process

Most of the existing composition approaches consider composition as a multistage process [7]. Composition process involves three stages namely, composition request formulation, composition plan generation and composition plan execution and monitoring as shown in Fig 1. At stage 1, a request to construct a customized composite service is formulated. In AI-Planning based approaches, a request to composite service is specified in terms of Input, Output, Pre-conditions and Effects (IOPE).

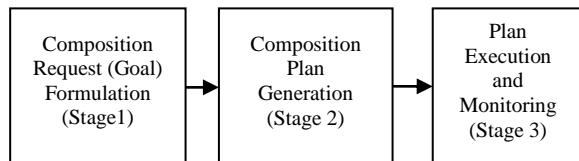


Fig. 1. Stages of Composition Process (Offline composition)

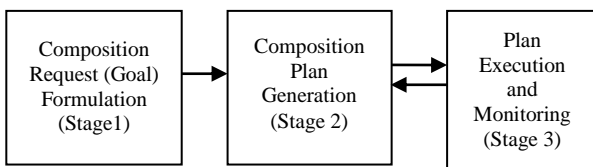


Fig. 2. Online Composition process

These approaches mainly concentrate on finding a composition plan for a given request. It is difficult for the non-technical end-users to formulate a composition request at too low level.

At stage 2, based on the composition request/goal a composition plan with abstract services description is generated. In AI-Planning based approaches, this is done in two ways namely offline composition and online composition [7]. In offline composition as shown in Fig.

1, composition plan is generated without monitoring the execution. A complete plan is generated before it gets executed in the execution environment. An abstract composition plan is constructed with abstract service descriptions. The abstract services are bound to the concrete services in the execution stage and the plan is executed. The plan is generated completely before its execution at one shot [8]. In online composition as shown in Figure 2, a composition plan is generated by monitoring the execution and in this, planning and execution are interleaved. A composition plan is generated partially and it is executed and then it produces the next partial plan and executes. This is repeated until the goal is reached.

At stage 3, the composition plan is executed and monitored. As the web services environment is highly dynamic and volatile many new services may appear and existing services may withdraw. In offline planning, there is a possibility that the composition plan may fail in the execution stage because concrete services that match the abstract service descriptions in the plan may become faulty or unavailable due to this dynamic nature of the service environment.

This is tackled by replacing the faulty one with an alternate service in the execution stage. If the alternate services are not available for replacement then the plan fails and a request for new composition plan is passed to stage 2 from stage 3. This is called

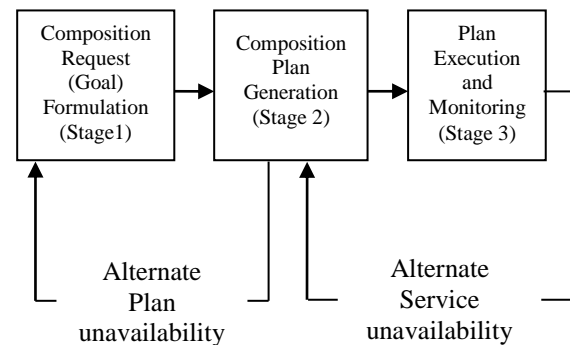


Fig. 3. Failure Handling in Composition process

Re-planning for the same composition request. If there exists no plan to reach the request then a request to refine the user needs and preferences is passed to stage 1 from stage 2. This is depicted in Fig. 3. In online composition approaches, dynamic re-planning is done. In dynamic re-planning, if the plan fails in stage 3 then stage 2 is informed to re-plan from the current state where the plan failed rather than from the scratch as done in the offline composition process. If there exists no plan to reach the request then a request to refine the user needs and preferences is passed to stage 1 from stage 2 [8]. This is depicted in Fig. 3.

B. Goal-Driven Composition Process

These approaches mainly concentrate on the end-user side allowing them to specify their needs, preferences and constraints in a natural way. The concept of goal is used to abstract the technical details from the end-user. At

stage 1 the Goal-driven approaches allow the specification of the composition request in terms of high-level goals. Goal ontology is used to transform a goal expressed in natural language to a formal composition goal that is understood by the next stage in the process. At stage 2, a composition plan with abstract services description is elicited from the formal goal with the help of domain ontology and services ontology. At stage 3, concrete services are discovered and matched to the abstract service descriptions in the composition plan. The composite service is now ready to be executed with concrete component services in it. This is depicted in Fig. 4.

IV. DYNAMIC SERVICE COMPOSITION ISSUES & SOLUTIONS

Though dynamic web service composition approaches construct the customized composite service based on the user request (goal) automatically and dynamically, they lack flexibility to deal with environment change. This is due to the reason that the environment where composition done is characterized by the following

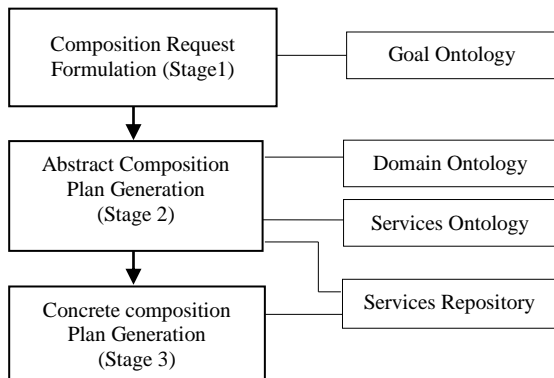


Fig. 4. Composition process stages in Goal-Driven Approaches

- Web services environment (run-time environment) is highly open [3] i.e. new services may be registered and existing services may be withdrawn (adding or deleting services). They are created and updated (changing QoS properties of service) on –the- fly.

Limitation: Web services composed through off-line planning may fail due to the above reason.

- Web services are huge in numbers [3].

Limitation: Scalability is the main issue for the composition process.

- Uncertain behavior of web services includes Non-deterministic (uncertainty in web service outcomes) and partial observability (internal state of a service is not seen) of the web services world [9].

Limitation: Composition process has to deal with incomplete information about the web services world.

- Changes in the needs of the end-users possibly mobile takes place [8].

Limitation: During the execution of composite service, context change is the major issue.

A. AI-Planning based Solution

Various AI-planning based approaches are available to tackle the above mentioned major issues and they are as follow

- The drawback of Off-line planning is tackled by adapting the composition plan to the runtime environment by Rebinding, Re-planning from scratch, Dynamic Re-planning (Re-planning from the current state) – BDI Agents are also used [10].
- Scalability issue is tackled by distributed planning and multi-agent systems (Team agents) [11].
- Uncertain behavior of web services is modeled by using a probabilistic variant of situation calculus and decision theoretic planners. Also by Extended Goal concept-Explicitly specifies the recovery goal [9].
- Context changes are tackled by dynamic re-planning.

B. Goal-Driven Solution

Existing goal-driven approaches mainly concentrate on allowing the systems to accept the user request from informal reality and transforming it into a formal one [12][13]. It enables the composition system to capture the user requirements expressed in real world terms (high-level) and formulates a high-level goal from which service composition is elicited automatically. In [14][15][16] Goal-driven service composition is discussed. Each approach serves a specific purpose. In order to capture and express complex user requirements, the concept of goal is used in all these goal-driven approaches and various goal models are proposed to capture the functional requirements which state the functionality of the system to be, non-functional requirements that state how to keep the system on track in case of anomalies and requirements that enable the system to be self-adaptive. The goal models specify the user criteria in terms of goals by making the non technology users to express their requirements in an easier way (natural language). A Goal based service ontology and domain ontology are used to transform the goal expressed in natural language into formal goal. Goal models are built by extracting the goal from the user request and decomposing it into sub-goals and arranging them in a structure [17][18]. Using the goal model, composition process is elicited based on the determined sub goals and their relationships. Therefore the solutions provided in Goal-driven approaches try to address the problem of composition request formulation for non-technical users and how to elicit composition of services from composition request and do not discuss about failure handling in all the stages of composition process.

V. RESEARCH GAPS

Both the approaches are based on the goal-driven architecture. In AI-Planning based approaches, given an initial state, an explicit goal definition and a set of possible state transitions (actions), a planner can automatically synthesize a plan of actions that may achieve the goal. When related to web services

composition web services are represented as actions and the desired composite service is represented by an explicit goal definition. Composition starts with explicit goal definition and Goal-oriented Action Planning (GOAP) takes place to choose right services for the composition [2][4][9]. There are many planners available in the literature to achieve composition. Each one address a specific factor which are listed below.

- Partial Observability - As the internal status and variables of web services cannot be accessed, they are partially observable.
- Incomplete initial state description-In Web services composition domain, a complete initial state description cannot be maintained.

- Non-Deterministic behavior of web services i.e. web services operations may fail during execution time or yield unexpected results.

Goal-driven approaches concentrate much on bridging the gap between user’s requirements expressed in real world business terms and services providers’ description of services in technical terms. Though these approaches give much weights to the composition request formulation to support non-technical users to state their request in a natural way, they too concentrate on elicitation of service composition process to some extent [14][15][16][17][18][19].

Table 1. Comparison of predominant dynamic Service composition approaches based on major dynamic service composition process stages with challenges and features in each stage

Approaches			Service composition request formulation		Service composition Plan Generation				Service composition Plan Failure Handling			
									Composition Plan (stage 2) failure		Composition Plan Execution (stage3) Failure	
			Origin	Model Availability	On-demand	On-the-fly	Automated		Reported	Handled	Reported	Handled
				Process creation	Service discovery							
AI Planning based	Offline Planners	M.Pistore & Traverso (2005) [9]	User request in Technical terms	Extended Goal-Model for expressing complex user requirements	Y	Y	Y	P	ND	ND	Y	Y
	Online Planners	BDI Agents & Team agents [10] [11]	ND	Desire / Goal base	ND	ND	N	N	ND	ND	Y	Y
Goal-Driven Approaches	Michael Stollberg et al.(2006) [5]		User request in Non-technical terms	GDO-Goal Description Ontology	Y	Y	Y	Y	ND	ND	ND	ND
	Maja Vukovic (2007) [8]		User request is captured thro his context (non-technical terms)	Context aware goal Model	Y	Y	Y	Y	Y	P	Y	Y
	Dmytro Zhovtobryukh (2007) [14][17]		User request in Non-technical terms	GDL4WSC	Y	Y	Y	Y	ND	ND	PD	PD
	Luiz Olavo Bonino da Silva Santos et al.(2011)[15][16]		User request in Non-technical terms	GSO-Goal Service Ontology	Y	Y	Y	Y	ND	ND	PD	PD
	Seheon Song et al.(2013)[20]		User request in Non-technical terms	Goal based service model	Y	Y	Y	Y	ND	ND	PD	PD

Y-Done; N-Not Done; P- Partially Done; PD-Partially Discussed; ND-Not Discussed.

An overview of the existing approaches with its composition process stages is given in table 1. Based on the details given in table 1, it is clear that AI Planning based approaches tried to focus on service composition generation, execution and failure handling for the user request specified in technical terms. These approaches try to generate on-demand, on-the-fly, automated service composition and handles the composition failures to

some extent. They attempt to tackle the dynamics of services environment during service composition plan generation stage. Whereas Goal-driven approaches try to focus on composition request formulations for user’s requests specified in non-technical terms and generate a composition plan which is ready to be implemented without stating anything about composition execution and failure handling. It shows that much of the work has been

done in generating an on-demand, on-the-fly, automated service composition rather than fault tolerant service composition. Although much work had been done to handle composition execution failure, only a few had tried to report and handle composition failure for a user request. This definition of dynamic service composition in these approaches seems to be too primitive and it is not sufficient to describe dynamism in all stages of services composition. The definition of dynamism is incomplete without reporting and handling the composition request failure automatically and on-the-fly. Therefore a detailed study on various dynamic service composition approaches is done and the definition of the term is found and tabulated in table 2. The tabulation shows that most of the research work uses the term dynamic to define the on-demand, on-the-fly, automated service composition process. It is observed from the table that the generic term dynamism is used in service composition plan generation stage and is extended with the term adaptability [20][21] in case of service composition execution failure handling. Only few approaches tried to extend the term dynamism

with the term flexibility in case of composition request failure handling.

VI. PROPOSED PROBLEM

It is understood that service composition is not flexible because of the distributed, openness and dynamic nature of services and it is error prone. Though few approaches had worked on flexible service composition, we find that there is still a plenty of room is available to improve the existing approaches and/or to propose a new approach for efficient flexible service composition. Therefore it is required to find an approach that could make the service composition highly dynamic, adaptive and flexible and could be termed as agile service composition. Our aim is to propose an approach that provides a high-level of dynamism in composition plan generation, adaptiveness in handling composition plan execution failure and flexibility in handling composition failure / composition request failure.

Table 2. Details of the terms dynamic, adaptive and flexible defined in service composition approaches

Existing Approaches	Defines	Parameters	Composition process stages	* Research strength
[8][9] [14][15] [16][17] [20]	Dynamic	On-demand On-the-fly Explicitly Automated	Composition Plan generation	High
[8][9] [10][11]	Adaptive or Reliable	On-the fly Explicitly Automated through Goal-driven (user requirements driven) or user Context-driven	Composition Execution Failure handling stage	Medium
[8]	Flexible	On-the fly Explicitly Automated	Composition Failure handling stage / Composition request failure handling	Low

*Research strength is measured by No. of Research Works done so far.

VII. PROPOSED SOLUTION

Hence our work is to propose an approach for agile service composition. Our approach proposes and builds a composition engine that captures the request from non-technical end users at high-level and builds an agile service composition. It uses the principle of abstraction and refinement to direct the service composition process end to end seamlessly from stage1 to stage3. A proposed model of Goal-Directed Orchestration engine is shown in Figure 5. The model consists of various components that each one does a specific function in the composition process.

The detailed description of the model is given below.

User Request: User's requirements such as functional requirements, preferences and constraints are obtained. These requirements are given as inputs to the engine.

Abstract Goal Formulator: User's requirements are captured through a goal model and allow the users to specify a top-level request or a low-level request. The top-level request is captured as a high-level goal and then decomposed into sub-goals which are arranged in a hierarchical tree structure. The high-level goal and sub-

goals are modeled as abstract goals. The abstract goal tree consists of abstract high-level goal at the top-level and means goals at the intermediate levels and end goals at the leaf levels.

Concrete goal Formulator: The concrete goal tree is generated by instantiating the abstract goal tree with concrete values that are provided by users' requirements and domain knowledge.

Abstract Task Formulator: It formulates the abstract task from the abstract goal by mapping the goal to task using goal-task ontology and domain knowledge.

Concrete Task Formulator: It formulates the concrete task by mapping the concrete goal to the corresponding concrete task and by instantiating the abstract task with the values obtained from users' requirements, domain knowledge and service discovery engine.

GDO Process: User request for service is given to the engine and the composition request generator generates an abstract goal tree and concrete goal tree. The Abstract goal tree is mapped to abstract task tree and the concrete goal tree is mapped to the concrete task tree and thereby a composition plan is generated with abstract service descriptions. If services are not available to construct a

composition plan then the goal cannot be achieved and it leads to goal failure. This failure is reported back to the composition requestor and an alternate sub-goal for the failed sub-goal is found and given to the composition plan generator. Thus our approach reports the failure of goal and handles it by alternate sub-goal.

VIII. USAGE SCENARIO

To illustrate how the proposed approach can handle composition request failure, this section introduces the following scenario in the scope of a travel trip planning domain [22]. Travel trip plans are built

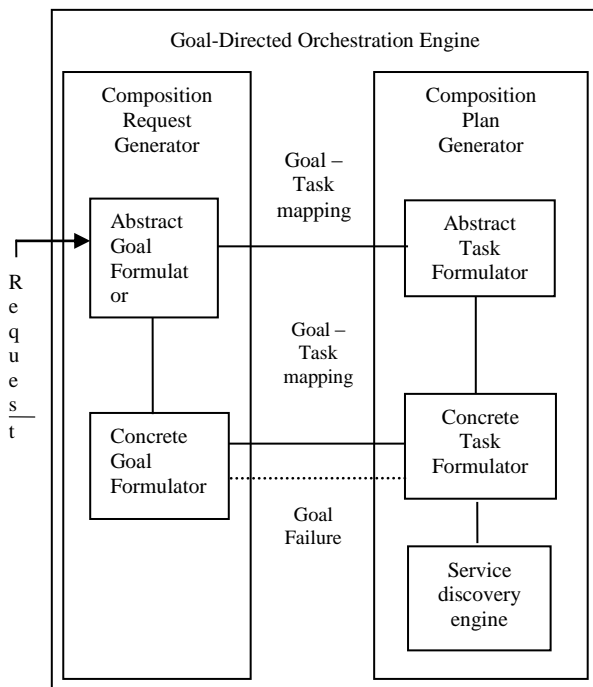


Fig. 5. A proposed model for GDO engine

based on varying travel requests. Eg. Users request may include flight booking, hotel booking and car rental while others may include only flight booking and hotel booking. Our approach tries to capture the user request at high-level with preferences and constraints. Eg. Plan a trip from source to destination with medium cost. The high-level request is transformed into an abstract goal tree with AND/OR sub-goals and alternate sub-goals. For eg. Plan a trip abstract goal is decomposed into plan a transport, plan an accommodation and plan a local

transport sub-goals. The top-level goal and its sub-goals are abstract and they are made concrete based on the source and destination of a trip. If the source is Pondicherry and Destination is Chennai then the abstract sub-goal Plan a transport is made concrete such as Plan a bus, Plan a train, Plan a flight, Plan a car. Therefore the sub-goals are made concrete and these provide a source for alternatives if the user request fails for one. Then the concrete sub-goal Plan a bus is mapped to a concrete task book a bus. The task is fulfilled by the service if it is available and the task fails if service is unavailable. Service unavailability leads to task failure which in turn leads to goal failure. Our approach handles this situation by replacing the sub-goal, plan a bus by an alternative sub-goal plan a train or plan a flight or plan a rental car. Thus our approach uses the principle of abstraction and refinement to handle the failure during composition plan generation.

IX. DISCUSSION

Our model provides a solution to agile service-composition by proposing a GDO engine. The proposed model in our approach is detailed above and it is compared with the models present in other approaches based on a set of features. The set of features used for comparison is not exhaustive but are necessary. They are found by having a detailed study on the composition request formulation and reformulation process. The identified features are abstraction level in user request, mapping of user request to goal, model for mapping, goal structure and goal reformulation process.

The features are explained in detail as below

Level of Abstraction in user request: A user can be allowed to specify his request at high-level (coarse-grained) or low-level (fine-grained). The level of abstraction in user request is mentioned as HIGH or LOW.

User request to goal mapping: This is required to identify which level of request is mapped to which type of goal. The values could be mentioned as LOW Level of request is mapped to task goal and HIGH level of request is mapped to Goal and sub-goals.

Mapping Process Model: This is required to identify whether the mapping is predefined and static or generated on-the-fly.

Table 3. Comparison of GDO approach with AI-Planning based and Goal-Driven approaches

Approaches	Level of Abstraction in user request	User request to goal mapping	Mapping Process Model	Goal Structure	Composition request / goal reformulation Process
AI Planning based	Fine grained (LOW)	Task goal	Predefined	Extended goal	Goal-driven
Goal-driven Approaches	Fine grained & Coarse grained	Task goal	Generated on the fly	Goal- means ends Hierarchy	Goal-driven & context driven
Goal-directed Approach	Fine grained & Coarse grained	Goal and means goals	Generate on the fly	Goal-means goal Hierarchy	Goal-directed

Goal Structure: Task goals or High-level goals (and/or its sub-goals) are arranged in a structure namely

sequential and hierarchical structure to capture and represent simple and complex user request respectively.

Composition request/goal reformulation Process: This deals with monitoring composition request/goal failure and reformulating the user request by user-driven (manual), Goal-driven (automated but predefined), user context driven (automated and dynamic by varying user context) and goal-directed (our approach). The comparison among the three approaches based on these features is listed in table 3.

The comparison shows that our approach handles composition request failure autonomously through the concept of goal and goal-directedness. Our approach is better than other approaches in satisfying coarse grained requests automatically and in handling request failure flexibly without the intervention of user.

X. CONCLUSION & FUTURE WORK

In this paper we have investigated the notion of dynamic service composition by focusing on the dynamic service composition process, approaches, issues and solutions. The research gaps are identified and the need for agile service composition approach is understood. Goal-directed approach has been proposed to provide agile service composition. Our approach is proposed with an aim that it provides a flexible response to changes. It includes an agile goal model which is highly resourceful and quickly adaptable to dynamic services environment. Thereby the service composition process is made flexible and adaptable to changing services world. An abstract goal model incorporated into the Goal-Directed orchestration engine has been proposed in this paper and formalizing this model is our future work. The significance of our work is shown by illustrating with a usage scenario trip planning in travel domain.

REFERENCES

- [1] Koehler, J., and B. Srivastava, "Web service composition – current Solutions and open problems.", *In Proceedings of the ICAPS 2003 Workshop on Planning for Web Services*, pp. 28–35, 2003.
- [2] V. Portchelvi, V. Prasanna Venkatesan, G. Shanmugasundaram, "Achieving Web services composition – A Survey", *In SAP publishing, Software Engineering*, 2012 2 (5). doi: 10.5923j.se.20120205.03
- [3] S. Dustdar, W. Schreiner, "A survey on web services composition", *Int. Journal of Web and Grid Service*, vol. 1, no.1, pp. 1–30, 2005.
- [4] Rao, J., and X. Su., "A survey of automated web service composition methods", *In Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition*, SWSWPC 2004.
- [5] Michael Stollberg et al., "Deliverable: WP3: Service Ontologies and Service Description D3.10 Goal Description Ontology", *Project Title: Data, Information and Process Integration with Semantic Web Services*, July 10, 2006.
- [6] Dominik Kuroпка, Anja Bog, and Mathias Weske, "Semantic Enterprise Services Platform: Motivation, Potential, Functionality and Application Scenarios", *EDOC*, page 253-264. IEEE Computer Society 2006.
- [7] Vikas Agarwal, Girish Chafle, Sumit Mittal, Biplav Srivastava, "Understanding Approaches for Web Service Composition and Execution", *in Proceedings of First Bangalalore Annual Compute conference*, Article no. 1, 2008.
- [8] Maja Vukovic, "Context-aware Service Composition", *Technical Report UCAM –CL-TR-700*, ISSN 1476-2986, 2007.
- [9] M. Trainotti, M. Pistore, G. Calabrese, G. Zacco, G. Lucchese, F. Barbon, P. Bertoli, and P. Traverso, "ASTRO: Supporting Composition and Execution of Web Services", *in Proc. of ICAPS*, 2005.
- [10] LI Huan, QIN Zheng, YU Fan, Qin Jun, YANG Bo, "Automatic Semantic Web Service Composition via Agent Intention Execution in AgentSpeak", *International Conference on Web Intelligence, IEEE/WIC/ACM*, 2007.
- [11] Xiacong Fan, Karthikeyan Umopathy, John Yen, and Sandeep Puro, "An Agent-Based Approach for Interleaved Composition and Execution of Web Services", *ER, volume 3288 of Lecture Notes in Computer Science*, page 582-595. Springer, (2004)
- [12] Kangkang Zhang, Qingzhong Li, Qi Sui, "A Goal-driven Approach of Service Composition for Pervasive Computing", *1st International Symposium on Pervasive Computing and Applications*, 2006.
- [13] Yoo-mi Park et al., "Semantic Web Service Processes Enabling Goal-Driven Application Creation", *15th International conference on Intelligence in Next generation Networks*, 2011
- [14] Dmytro Zhovtobryukh, "A Petri Net- based Approach for Automated Goal-Driven Web Service Composition", *SIMULATION*, Vol. 83, No. 1, 2007.
- [15] Luiz Olavo Bonino da Silva Santos et al., "Towards a Goal-Based Service Framework for Dynamic Service Discovery and Composition", *Sixth International conference on Information Technology: New Generations*, 2009.
- [16] Luiz Olavo Bonino da Silva Santos et al., "Towards a Conceptual Framework to Support Dynamic Service Provisioning for Non-Technical Service Clients", *Journal of Software*, Vol.6, No. 4, April 2011.
- [17] Manshan Lin et al., "Goal Description Language for Semantic Web Service Automatic Composition", *Proceedings of the 2005 Symposium on Applications and the Internet*, SAINT'05.
- [18] Tomas Olsson et al., "Goal Refinement for Automated Service Discovery", *Service Computation 2011: The Third International Conferences on Advanced service Computing 2011*
- [19] J.L. Hainaut et al., "GOORE: Goal-Oriented and Ontology Driven Requirements Elicitation Method", *ER Workshops 2007, Springer-Verlag Berlin Heidelberg, LNCS 4802*, pp 225-234, 2007.
- [20] Seheon Song, Seok-Won Lee, "A goal-driven approach for adaptive service composition using planning", *Mathematical and Computer Modeling*, Volume 58, Issues 1–2, July 2013, Pages 261-273, 2013.
- [21] Luciano Baresi and Liliana Pasquale, "Adaptive Goals for Self-Adaptive Service Composition", *IEEE conference on Web services*, 2010.
- [22] Shirin Shorabhi, "Web Service Composition via Generic Procedures and Customizing User Preferences", *International Semantic Web Conference, volume 4273 of Lecture Notes in Computer Science, page 597-611. Springer, 2006.*

Authors' Profiles



Portchelvi Venkatachalam born in Puducherry, received the B.Tech Degree in Computer Science Engineering from the Pondicherry Engineering College in 1997 and M.Tech Degree (Gold Medalist) in Computer Science Engineering from Pondicherry University, Pondicherry, India in 2007. She is currently working towards the Ph.D degree at the Department of Banking Technology, Pondicherry University. Her research Interests include Object-Oriented Software Engineering, Software Architecture and Service Oriented Architecture. She has published papers in Indian National, International Conferences and in Microsoft Digital Network Library. She has co-authored a book on Services Composition and Orchestration: Concepts and Approaches – Goal Directed Approach to Services Orchestration.



Prasanna Venkatesan is an associate professor in Banking Technology, Pondicherry University. His interest includes Software Engineering and Banking Technology. He has co-authored a book on Services Composition and Orchestration: Concepts and Approaches – Goal Directed Approach to Services Orchestration.

How to cite this paper: V. Portchelvi, V. Prasanna Venkatesan, "A Goal-Directed Orchestration Approach for Agile Service Composition", *International Journal of Information Technology and Computer Science (IJITCS)*, vol.7, no.3, pp.60-67, 2015. DOI: 10.5815/ijitcs.2015.03.09