A Goal Model for Adaptive Service Composition

V.Portchelvi
Research Scholar
Dept. of Banking Technology
Pondicherry University
Pondicherry,
South India

V.Prasanna Venkatesan
Associate Professor
Dept. of Banking Technology
Pondicherry University
Pondicherry,
South India

ABSTRACT

Achieving fully-automated service composition is a major requirement in SOA based systems. They try to satisfy the user request to composite service automatically by composing the services at one stage and executing the composed service at the next stage. User request has been transformed formally into the concept of goal and goal models and the success of the goal determines the satisfaction of user request. Goal failure returns no composite service and the user request is not satisfied. The success/failure of goal is based on the ability of the composition process to automatically adapt itself to the dynamic and complex services environment. AI-planning based and Goal-driven approaches, based on the concept of goal and goal models, provide effective solutions to develop fully-automated composition. They semi-automatically adapt the composition to dynamic and complex services domain. Semi-automated adaptive composition approaches try to handle composition (plan) failure and do not handle composition request (goal) failure. Only few approaches try to handle goal failure and that too manually. The concept of goal and goal models in these approaches are explored and research gaps are identified to improve these models so that they aid in producing highly adaptive composition. The aim of this work is to make explicit that the decoupling of Goal Failure and Plan Failure and handling them automatically is essential for full automated service composition. This is made possible by proposing a goal model that can provide more number of alternates to failed goal. The proposed goal model is well explained with an application example taken from Online Travelling Domain. Then the model is compared with the existing models.

General Terms

Software Architecture, Software Engineering, SOA.

Keywords

Automated composition, Dynamic service composition, Adaptive, Goal-Driven, Goal-Model.

1. INTRODUCTION

Service-Oriented Architecture (SOA) is an architectural style that bridges the gap between the business world and IT world. It creates a more powerful and valuable relationship between the two world. SOA allows today’s enterprise’s IT architecture to build service-based business solutions with services as the fundamental building blocks. Every individual business service encapsulates particular business functionality and is reusable. One of the most important features of SOA is to build a new business solution by combining the existing business services. This process of combining the business services into a composite service, coordinated by business process is called service composition. Service composition enables an enterprise to rapidly build a high level solution by recursively combining the composite service with existing business services. To realize SOA, Web services provide a standardized way to expose business functionality as service; Enterprise Service Bus (ESB) provides infrastructure support for services interaction and management and Business Process Execution Language (BPEL) for composing services into business process [1][2][3]. As the business world is dynamic, SOA‘s major promise is to provide a more flexible and adaptable business solution. Therefore an agile service composition approach is required to compose business services in the dynamic business world. Service composition is a process of combining two or more existing services into a composite service. Using pre-existing, tested services enhances service reusability and thereby decreases the effort to develop new solutions from the scratch. This process is essential in a situation where an individual service is not able to provide the required functionality. Composition process (system) is sequential and multistage. The composition process starts with a request to a composite service (specification phase) and based on this request a workflow plan to combine services is constructed automatically (Planning phase). Then the services that are to be combined are discovered (Discovery phase) and finally the services are orchestrated (Execution and Monitoring phase) to provide the required composite service [4]. The challenge is to find an approach that can handle the entire composition process automatically without any human intervention, ideally in all possible situations. In a fully automated composition process, the task of a human is to specify the request to a composite service and the process automatically produces the required composite service. Since services in SOA are distributed, there is a possibility that failure such as services unavailability and service changes may occur during the composition process. Composition modeling should consider the dynamism in the services environment. Fully automated service composition process is necessary to tackle the above issue.

In the literature various composition approaches are available to achieve a fully automated service composition. AI-planning based and Goal-driven approaches based on the concept of goal and goal models [7] provide effective solution to achieve service composition. But these approaches are limited in handling failures in composition process. Therefore an approach for adapting the composition to dynamic and complex services environment is required which is termed as adaptive service composition approaches. Many research works are done in achieving adaptive service composition [21] through the concept of goal and goal models. Goal models in the existing adaptive composition approaches try to overcome the above limitation semi-automatically. Hence fully automated service composition is not achieved. The aim of this paper is to study about the goal models in these approaches and to identify the research gaps that limit the achievement of adaptiveness in composition process. This is detailed in sections 2 and 3. The gaps are then bridged by proposing a goal model and applying it to online travel domain to demonstrate that the model aids in achieving the adaptive service composition. This is detailed in section 4. To show that the proposed model is qualitatively better than the existing models a comparison is done and shown in section 5. Section 6 discusses about the conclusion and our future work.

2. RELATED WORKS

A Research on service composition approaches is done both in the industry and in the Academia. In Industry, composition approaches adopt web services standards for service interactions and BPEL4WS and WSCI for composing these web services into business process. This leads to a syntactic based composition as the connection between web services are established based on the service input and output. The academic research community utilizes semantic web standards such as OWL-s and WSMO to perform service composition. This leads to a semantic based composition [6] as the connection between web services are established based on the service input, output, pre-conditions and effects. The services in the semantic web are semantically annotated so that automation of service discovery, composition and execution is achieved. Till now, automated service composition approaches surveyed from the literature, try to build a composite service on the fly based on the varying user request. This is termed as the dynamic service composition in the literature.

2.1 Dynamic Service Composition (DSC) Approaches

Dynamic service composition approaches build composite services dynamically for each varying user request. On demand, these approaches satisfy user requests by combining existing basic services into composite service on the fly. In the literature, most of the approaches to dynamic service composition realize service composition through a standard based technology called web services. The term web services composition is used to describe the composition of web services in a process flow. Business world tries to integrate their application and automate B2B interactions through web services composition by combining web services that belong to various business organizations. The main challenge addressed by DSC approaches is web services domain complexity. The complexity in this domain can be characterized as Exploratory, Volatile, Uncertain/Non-deterministic, Partially observable and Heterogeneity [3]. The other issues related to web services composition are composition performance, adaptability, failure resilience, composition correctness, optimization and execution. All these issues are addressed by the existing DSC approaches and thereby the literature has several and various DSC approaches, each tackling a specific issue. There is no complete, comprehensive approach that addresses all these issues.

2.1.1 AI Planning based Composition Techniques & Process

In order to achieve dynamic service composition, existing approaches rely on the automated problem solving techniques in Artificial Intelligence (AI). Different branches of AI research develop frameworks for describing problems and resources along with technologies for automated problem resolution and resource usage. AI Planning techniques, the subfield of AI are extensively used to achieve DSC. In planning domain, planning problems are logically structured as states, actions and goals. Planners are problem solving algorithms that find a sequence of actions that will achieve a goal. Various planners are available for different kinds of problem. AI Planning based DSC approaches take composition problem as a planning problem and tries to find a composition plan. These approaches consider the user requests as goal and perform a Goal-oriented Action Planning (GOAP) on web services domain to produce a composition plan that satisfies the user request. Since Web services environment is partially observable, non-deterministic and dynamic, the planner has to work in a complex environment. Various AI planning based approaches for Dynamic Web services composition are available and they can be broadly classified into offline composition approaches and online composition approaches [5].

Offline composition approaches attempt to achieve on-demand, automated composition in 3 different ways namely monolithic composition and execution, staged composition and execution and template based service composition and execution. Monolithic approaches consider composition as one shot plan synthesis problem. It assumes that all the generated plans will execute successfully without any failure. Existing approaches such as SHOP2 and GOLOG [22] has produced composition plan with an assumption that the plan never fails during execution stage. In order to overcome the drawback, various planning methods are devised and used in web services composition approaches. One such planning method is Execution Monitoring and Replanning method which attempts to solve the plan execution failure. At execution stage, in case a service in the composition fails due to its unavailability during execution time, execution monitor detects the service failure and the composition execution is halted and the not executed part of the composition plan is handed over to the planner and the planner replans for the current world situation in order to satisfy the original user specified request. If no such composition plan is available at planning stage this approach terminates the composition process and declares goal failure.

ASTRO [9] is a framework supporting automated service composition, monitoring and execution. This framework is based on the concept of planning as model checking. This framework views the composition problem as a single shot plan synthesis problem and handles plan failure at execution time by the execution monitoring and replanning method. In the staged approach planning and execution stages are sequenced. Composition process involves building a composition plan with abstract service descriptions at stage 1 and binds concrete services at stage 2. The planner does not monitor the execution. It can only return service composition upon user request. Execution failure occurs when no concrete services matching the abstract service descriptions in the composition plan could be found. [] is a context-aware framework which performs composition in stages and handles plan failure at each stage of composition process namely composition stage and execution stage.

In On-line service Composition approaches the composition problem is not viewed as a single shot plan synthesis as in off-line approaches. Instead the process of composition plan generation and execution is interleaved which means that composition planning and execution go hand in hand [10][11]. These approaches employ a planner that monitors the execution environment while planning. It does incremental planning in which it plans for a fixed time period and executes the plan. After plan execution it generates the plan for next time period. This process iterates until the goal state is reached. It provides a high level of adaptivity during execution stage. This approach faces scalability issue and is ideal if the search space is small. Failure handling in this approach requires a rollback of the services executed [5].

2.1.2 Goal-driven Service Composition Techniques & Process

Another automated problem solving technique that has gained importance in achieving services composition is Goal-driven architecture. The composition approaches based on this architecture allows the users to express their complex requirements at high-level and the user request is transformed into a formally described top-level goal[17][18]. With this top-level goal, a composition plan is generated automatically by an intelligent mechanism based on the decomposition of the top-level goal into subgoals and the relationships among these subgoals [14]. The concept of goal definition, the automated formal high-level goal formulation from informal user request and the automated composition plan generation by decomposing this high-level goal are the key techniques/steps in the Goal-driven dynamic service composition approaches. These approaches try to concentrate much on allowing the system to accept high-level informal user request and automated composition plan generation for the user request using the concept of goal and goal formulation. The concept of goal provides an abstraction so that the user can express their request to composition in an informal way i.e. in non-technical terms. Goal formulation involves decomposing the top-level goal into a process of goals which guides the planner or reasoner to construct the appropriate service composition plan. Concrete services are discovered to populate the plan with concrete functionality so that the composition plan is ready for execution.

2.2 Adaptive Service Composition Approaches (ASC)

The focus of existing DSC approaches is to make the composition system fully automated in constructing on-demand, on-the-fly composition plan (termed as dynamic service composition in the literature) but still it could not be achieved. This is due to the dynamism and openness of web services, services may become unavailable and services may keep changing by updating their Qos attributes. In such complex and dynamic web service environment, there is a possibility of plan failure at execution stage. All these approaches recover plan failure or adapt the plan according to the dynamism in the environment from the perspective of services unavailability and deal with the issues in replacing the unavailable services with alternate services that are equally capable. If alternate services are unavailable then these approaches finds alternate plans. If alternate plans are exhausted without satisfying the goal then goal failure occurs and these approaches returns no response to user request and hence no composite service is returned. Goal failure [8] is reported in some of these approaches but not handled. Composition adaptation is done by re-binding and re-composition. The execution environment provides event-driven behavior that enables the composition system to react quickly to changes in the web services domain by adapting its composition plan or by creating a fresh plan for the same planning goal.

3. LIMITATIONS

The success or failure of a goal depends on the availability or unavailability of the sequences of actions that can satisfy the goal. It could be inferred that the approaches discussed above are trying to establish a relationship between the dynamism in complex web services environment during composition execution stage and the improvement in goal satisfaction. Existing approaches view composition goal as a request for composite service and composition goal is represented as conjunction of goal states and some approaches like ASTRO try to satisfy all the goal states to satisfy the composition goal. If any one of these goal states fails, the entire composition goal fails and returns no composite service. The satisfaction of user request (goal) in these approaches is all or none. The other approaches aims in satisfying some of the goal states and leaves the failed ones. Their objective is to provide a composite service that partially satisfies the composition goal rather than no composite service at all. Goal Morph [8] has been devised to address the above problem by partial satisfaction planning method. They try to satisfy the goal fully or partially by on the fly composition execution failure recovery. Therefore a high-level of self-adaptability of service composition is required to achieve a full automated service composition that in turn improves the goal satisfaction without any human intervention. Though these approaches try to rebind and recompose (replan) to handle composition execution stage failure they lack methods/techniques to handle goal failure.

4. PROPOSED SOLUTION

Our aim is to propose a goal model to handle the goal failure and to aid in achieving a high-level adaptable service composition. To propose this goal model a need to find the research gaps in the composition process arise and are identified through a study on failure handling in the composition process in the existing approaches. We identified that the goal failure is not reported and handled automatically in these approaches and this motivated us to propose a goal model to handle goal failure automatically. The Goal model is described in sections 4.2 and 4.3.

4.1 Motivation

Since much work has been done in constructing fresh plans and alternate plans to adapt to changing services domain and to satisfy the user request, and it is found from the above survey in section 2 of this paper that no research gap is available in this planning stage of composition process, it paved a way to find the research gaps in the goal definition and goal formulation stages which may lead to find some solution to improve goal satisfaction through high-level adaptive service composition in the dynamic domain of services environment. This motivated us to explore the concept of goal, its definition, its formulation and its usage in service composition approaches. A study has been done on the existing approaches based on the below mentioned objectives and the results are tabulated in Table 1.

- What is the goal definition given in the models?
- What is the purpose of the goal model? (for DSC or ASC)
- What is the nature of the defined Goal? (Domain specific, Application specific, Generic)
- What types of goal attributes are mentioned? (general attributes such as Name, Description, Priority etc. and application specific attributes i.e. attributes related to problem domain)

- What types of relationships exist among the goals? (Modeling structure)

From the above table 1 it is clear that only few goal models [20][21] support adaptive service composition and the rest focuses on dynamic service discovery and composition. Only [21] discusses about modeling adaptation strategies at the requirements elicitation level and the adopting them during runtime. The above mentioned goal model is an extended Goal Oriented Requirements Engineering (GORE) KAOS model [21]. Many of them do not discuss about the goal success rate. Their aim is to bridge the gap between informal user request and the services implementation. Therefore a goal model is required to improve the goal success rate and provide high-level adaptive approach. A goal model has been proposed and based on which adaptive service composition has to be done.

4.2 Goal Model

The goal model is proposed based on the idea of goal change/goal formulation inherited from game theory. The goals of the game change as the game state changes. Eg. In chess, if a win is infeasible then an attempt to gain a draw is made. Our model handles the goal failure by providing alternate goal states for the failed ones. It aids the approach based on it to do continuous planning for new formulated goal in order to provide a real adaptation and flexibility. Therefore a composition approach based on the proposed model provides and shows a real adaptation to changes in dynamic web services environment and thereby the success rate of goal satisfaction is improved. The proposed goal model provides goal knowledge for the real adaptivity. It makes the system to be aware of its goal knowledge to provide a real adaptivity to dynamic environment.

4.3 Goal Tree

The goal model has been designed based on the design principle of Object-oriented Design, namely **Decomposition and Abstraction.** Decomposition principle is used to decompose a large problem into sub problems. Each sub problem is at the same level of detail, can be solved independently and can be combined to solve the original problem. Based on the above definition our goal model is designed to specify a high-level goal which is decomposed into subgoals and a hierarchical ordering of the subgoals is done. Goal tree structure is used to represent the model. In this the high-level goal (problem) is decomposed into one or more subgoals (sub problems) and each subgoal is decomposed further into one or more lower level subgoals. There are two types of relationship among the subgoals namely is-a and part- of. The goal/subgoals are defined with abstract property and the is-a relationship is used to generate alternate subgoals to each abstract subgoal at each level to handle goal failure at each level. This aids in building a high-level adaptive service composition. The principle of abstraction by specification is followed which allows each goal/sub goal in the goal model to specify what is to be achieved and abstracts away/ignores how it is done. The goal model captures the users request in terms of what is to be achieved and modeled as goal tree.

4.4 Task Tree

The means to achieve the goal has been termed as tasks and they are modeled as task tree. The task tree is used to represent how to achieve the user needs. Each node in the tree is an abstract subtask for which many instance subtasks and part-of subtasks can be created. The task tree depicts parent/child relationship and preferences and constraints among the nodes. The plan for composing the services is now ready in the form of task tree with each instance subtask has all the details to discover concrete services. The separation of goal and task tree with their meta-model representation is shown in Figure 1. The proposed goal model is generic and can be applied to different domains where it is suitable.

4.5 Application Example

In order to demonstrate that our goal model can be applied to an application scenario, we have chosen Online Travelling Domain for discussion. In today’s Internet world people prefer to book their trips online and they have multiple needs such as planning for transport, accommodation, etc. in their mind to make their trip pleasant and comfortable. The needs of the user can be catered by different service providers available in the web. Dynamic service composition approaches can give an automated, one-stop shop solution to this trip booking scenario. Based on our proposed goal model the DSC approach can provide an automated high-level self--adaptable service composition in this scenario and is shown in Figure 2 and Figure 3. The goal model and task model shows/exhibits all possible plans available for booking a trip in an exhaustive manner to make the user’s trip plan possible by all/any means.

User request to obtain a trip plan from Pondy to Chennai along with his preferences and constraints are taken as input and based on our goal model a goal tree is generated with the top-level goal as ‘Achieve Trip plan from Pondy to Chennai’. The subgoals for this top-level goal are ‘Achieve transport mode plan from Pondy to Chennai’ and/or ‘Achieve Accommodation mode Plan from Pondy to Chennai’. Each of the above mentioned subgoals have instance-of subgoals which are related with ‘or’ relationship. If the subgoal ‘Achieve transport mode plan from Pondy to Chennai’ is considered then its instances are ‘Achieve Bus plan from Pondy to Chennai’, ‘Achieve Train plan from Pondy to Chennai’ and ‘Achieve Rented car from Pondy to Chennai’ and the relationship among them are OR. Similarly if we consider the subgoal ‘Achieve transport mode plan from Pondy to Bangalore’ then its instances are ’Achieve Bus Plan from Pondy to Bangalore’, ‘Achieve Train Plan from Pondy to Bangalore’, ‘Achieve Rented Car from Pondy to Bangalore’ and ‘Achieve Flight plan from Pondy to Bangalore’. Users’ preferences and constraints are used to decide the AND/OR relationship among the subgoals and the ordering of the subgoals’ instances.

This shows that the goal model aids in formulating the user preferred dynamic goal tree for different source and destination. Also from the Figure 2 it is evident that the user is able to plan his trip as a whole (one-stop shop trip planning) i.e. he can plan for his transport, accommodation, local attraction arrangements etc. through a single request of ‘To obtain a trip plan’.

5. DISCUSSION

Many research works has been done in adaptive service composition area in the past decade. Only few researchers proposed goal models for achieving high-level adaptive service composition. This section focuses on the details of those models and the comparison of our model with them. The purpose of the goal models is to aid in capturing complex composition requirements and to help in building self -

adaptive composition which could adapt more flexibly and dynamically to the changes in the complex services environment and changes in user requirements. Existing goal models and the proposed goal model are compared based on the following parameters shown in Table 2. Also many other parameters aid in understanding the goal models better and are listed below.

- How the goal is formulated? (User given goal (manual), User assisted goal (semi-automated, System generated goal (automated))
- Does the goal model follow any lifecycle?
- Where and which stages of composition, goals can be used?
- Whether goals can be changed /extended/reused for capturing new/changing complex user requirements?
- Whether goals can be used by inference mechanism to derive a composition plan?
- Whether goals can aid in Plan failure handling during execution?
- Whether the success and failure of goal can be used for future i.e. for learning?

From the table 2, we can infer that the existing goal models aid in achieving dynamic service discovery and composition rather than adaptive service composition. The proposed goal model facilitates adaptiveness by providing goal based abstraction rather than action based abstraction.

6. CONCLUSION

In this paper we presented various existing dynamic composition approaches and the limitations in those approaches were identified. The requirement for adaptive service composition is stressed for achieving fully-automated service composition and to improve the success rate of goal success. To find the research gap where it could be improved to find solution, a study on existing goal models were done and the results were tabulated. From the study it was inferred that the goal model can aid in adaptiveness of service composition. Based on this inference a new goal model was proposed and a qualitative evaluation was done with the existing models. From the comparison the proposed model, exhibits a high scope for aiding adaptiveness in service composition. The future work is to propose an adaptive service composition approach based on this goal model and to implement it through formal abstract state space machines.

7. 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”, SAP publishing,Software Engineering. 2012 2 (5). doi: 10.5923/j.se.20120205.03

[3] S. Dustdar, W. Schreiner, “A survey on web services composition”, Int. Journal of Web and Grid Services, 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] Vikas Agarwal, Girish Chafle, Sumit Mittal, Biplav Srivast ava, “Understanding Approaches for Web Service Composition and Execution”, in Proceedings of First Bangalaore Annual Compute conference, Article no. 1, 2008.

[6] Dominik Kuropka, Anja Bog, and Mathias Weske,” Semantic Enterprise Services Platform: Motivation, Potential, Functionality and Application Scenarios EDOC, page 253-264. IEEE Computer Society 2006.

[7] 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.

[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] Xiaocong Fan, Karthikeyan Umapathy, John Yen, and Sandeep Purao, "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”, (SAINT’05) Proceedings of the 2005 Symposium on Applications and the Internet.

[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", 2007.

Table 1. Results of a study on Goal in the Existing Goal models of Service Composition Approaches

	Dmytro Zhovtobryukh [2007]	Luciano Baresi and Liliana Pasquale [2010]	Luiz Olavo Bonino da Silva Santos et al [2011]	Seheon Song, Seok-Won Lee [2013]
Purpose	Provides necessary details for Dynamic Service discovery & composition	Provides necessary details for Dynamic & Adaptive Service Composition	Provides necessary details for Dynamic service discovery and composition	Provides multiple goals to HTN planner to generate Dynamic Service composition plan
Is Concept of Goal Defined	Yes	Yes	Yes	Yes
Is Nature of Goal Stated	Yes	Yes	Partial	Partial
Is Goal attributes mentioned	No	Partial	No	No
Does Modeling structure support adaptation strategies	Partial	Yes	Partial	Partial

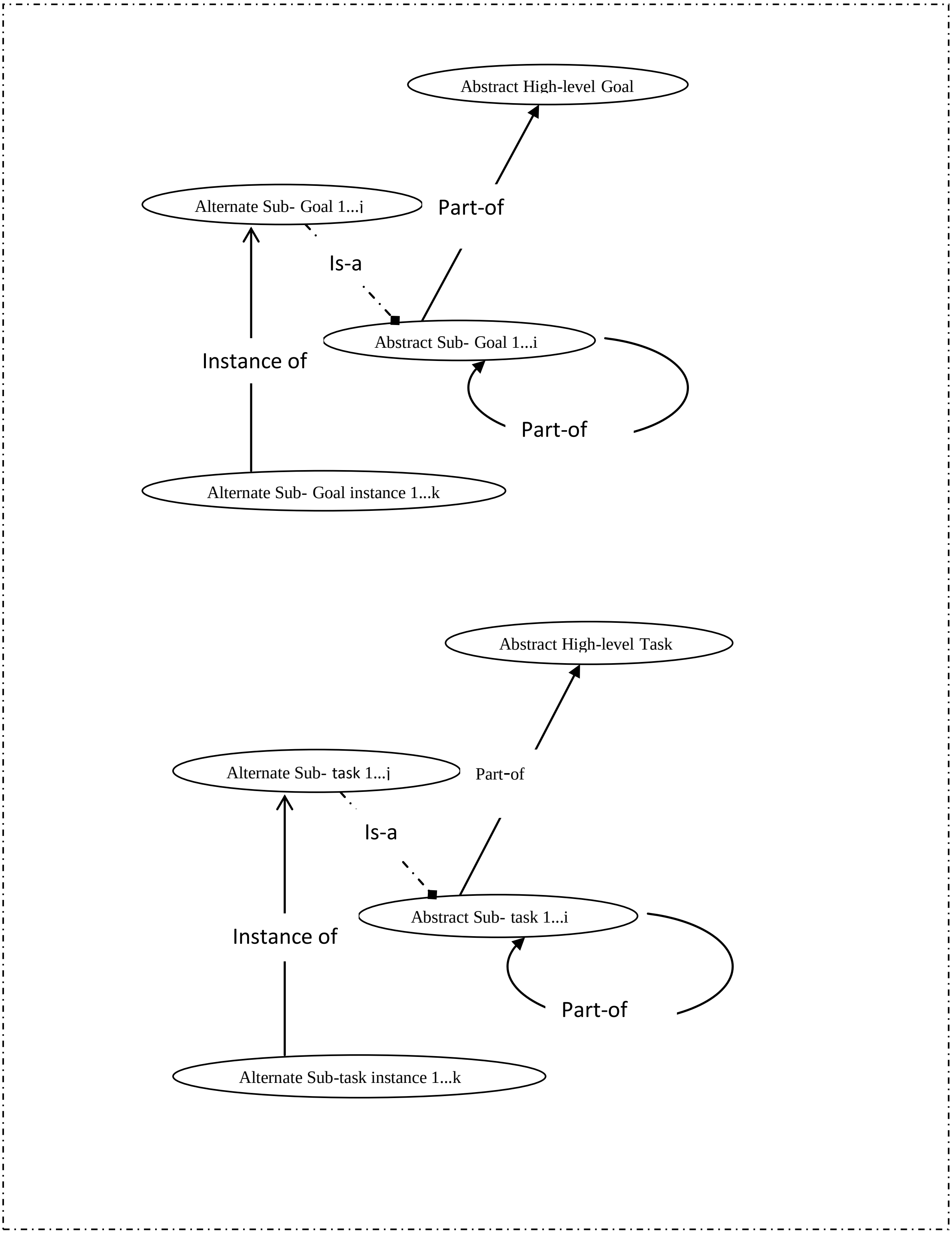

Fig.1: Separation of Goal and Task Tree and their meta-model representation

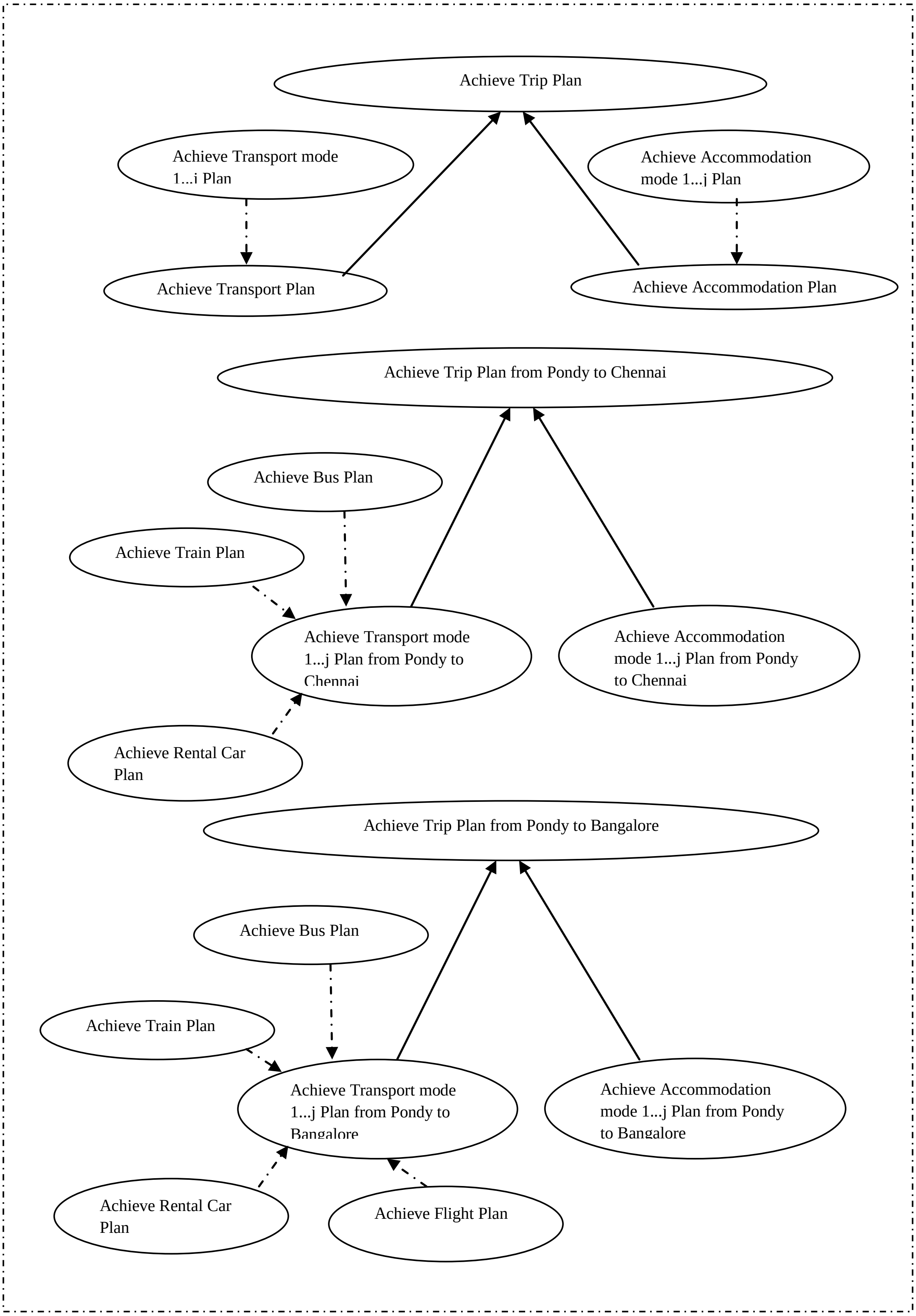
Achieve Trip Plan
Achieve Transport mode 1...i Plan
Achieve Accommodation mode 1...j Plan
Achieve Transport Plan
Achieve Accommodation Plan
Achieve Trip Plan from Pondy to Chennai
Achieve Bus Plan
Achieve Train Plan
Achieve Transport mode 1...j Plan from Pondy to Chennai
Achieve Accommodation mode 1...j Plan from Pondy to Chennai
Achieve Rental Car Plan
Achieve Trip Plan from Pondy to Bangalore
Achieve Bus Plan
Achieve Train Plan
Achieve Transport mode 1...j Plan from Pondy to Bangalore
Achieve Accommodation mode 1...j Plan from Pondy to Bangalore
Achieve Rental Car Plan
Achieve Flight Plan

Fig: 2: Domain and Instance model of Goal (Virtual Travel Agency)

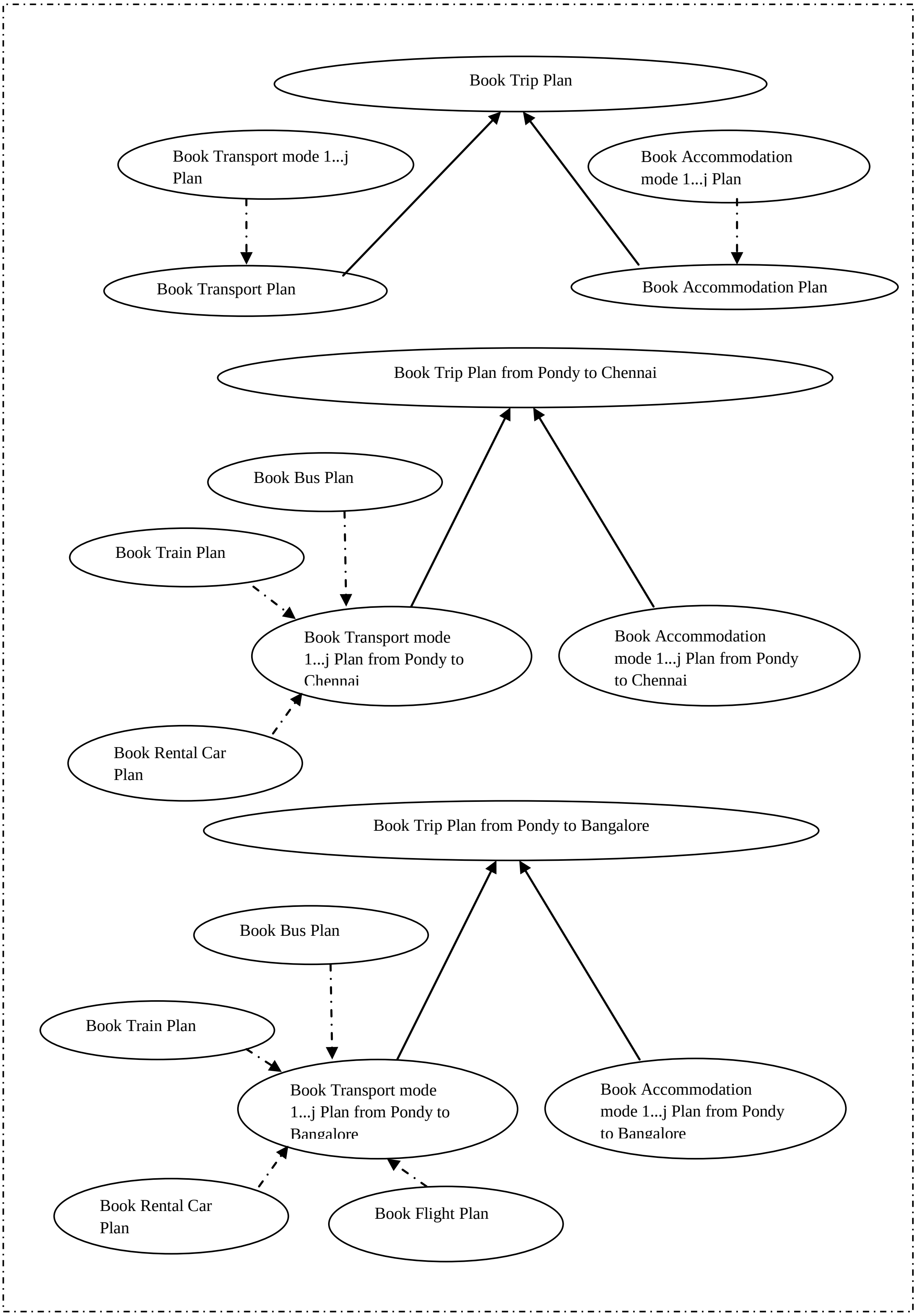
Book Trip Plan
Book Transport mode 1...j Plan
Book Accommodation mode 1...j Plan
Book Transport Plan
Book Accommodation Plan
Book Trip Plan from Pondy to Chennai
Book Bus Plan
Book Train Plan
Book Transport mode 1...j Plan from Pondy to Chennai
Book Accommodation mode 1...j Plan from Pondy to Chennai
Book Rental Car Plan
Book Trip Plan from Pondy to Bangalore
Book Bus Plan
Book Train Plan
Book Transport mode 1...j Plan from Pondy to Bangalore
Book Accommodation mode 1...j Plan from Pondy to Bangalore
Book Rental Car Plan
Book Flight Plan

Fig.3: Domain and instances Model of Task

Table 2: Comparison of the goal models

Goal Models Parameters	Luciano Baresi and Liliana Pasquale [2010] [b]	Seheon Song, Seok-Won Lee [2013] [d]	Proposed goal model
Goal reduction technique	Procedural goals	Procedural goals	Declarative
Goal State Representation	Perform goals	Perform goals	Achieve goals
Goal-action synchronization	Top-level Achieve goal to action Tree mapping	Leaf-level achieve goal is mapped to Composite task	Each achieve goal in each level is mapped to each tasks in each level of the task tree
Failure recovery	Action-level (Action-based)	Action-level (Action-based)	Action-level (Action-based) & Goal-level (Goal-based)
Types of Alternatives	Action alternatives	Action alternatives	Action alternatives and goal alternatives
Automated ordering of subgoals	No	No	Yes based on user preferences and constraints
Levels where alternatives are available	Leaf level	Leaf level	At each intermediate and leaf level