Intelligent Task Resource Planning Based on Adaptive Dynamic Resource Composition Model

Linyun Lu¹, Xinhuai Tang²,*
¹,²School of Software Engineering, Shanghai Jiao Tong University, Shanghai, China
*Corresponding author e-mail: tang-xh@cs.sjtu.edu.cn

Abstract. With the rapid development of information technology, various resources, including computing, network, storage and application, generally present the trend of servitization. In order to plan these resources dynamically in a short time according to the task requirement and allocate them properly, this paper propose an adaptive dynamic resource composition model which can realize fast searching and matching of resources through automatic classification of resources based on type, Hierarchical Task Network (HTN) planning to implement the task planning and resource allocation algorithm to implement the resource allocation. We design a task resource intelligent planning system with a specific scenario to verify our theory and method.

1. Introduction
In many situations, we should find the most suitable resources for an emergent task like joint operations and dam emergency. Many tasks may be complex, dynamic, diverse and emergent [1] for we to allocate resources. At present, the market still lacks the system which can finish the task decomposition according to assigned tasks. So we propose an adaptive dynamic resource composition model, the intelligent Hierarchical Task Network (HTN) planning and Multi-dimensional Dynamic List Scheduling (MDLS) algorithm. At the end of the paper, we use an example to verify our theory and method.

2. Ontology-based Semantic Representation Model
We propose a semantic model of resource domain and type based on ontology, which combines ontology technology and resource domain, describing resources with the same or similar functions. It uses ontology reasoning technology to realize automatic classification of resources and then realizes the self-evolution of its classification structure.

Firstly, we talk about the Ontology Language for Services (OWL-S), which used to model the resource by semantic ontology, describing resource interface, structure, message and so on at the semantic level, aiming to support key processes such as automatic resource discovery.

2.1. A Semantic Representation Model
OWL-S is based on ontology that can make resource description understandable. The description of the resource by OWL-S is similar to the structure of HTN planning. As a result, it makes it possible for us to implement task planning for we can transform them easily. By using this resource language, we can describe our resource including functional and non-functional attributes(QoS). For example, a computing resource has an attribute "disk " as show in Fig. 1:
2.2. Domain and Type
As we have said above, the method that extracting resource type from resources and the service type is divided into service domain by function for management is used to manage the resources in a growing number.

Reference [2] proposes the concept of service domain in order to quickly find alternative Web services when Web services fail. The service domain is defined as a middleware solution of service broker and management to solve the change of service domain structure. However, the service described by WSDL does not have semantic features. As a result, the creation, modification and other maintenance operations of the service domain still need to be done manually.

So we can use the idea that combine domain with OWL-S to realize our resource semantic model.

Also, the concept of type is different from that of domain, but their functions are similar. They are based on function abstraction of resources. The difference is that the resource type is just a description of the resource function, not dynamic, and unable to cope with changes in the resource type.

2.3. Hierarchical Task Network Planning
With the emergence of OWL-S, the dynamic service composition method based on Intelligent planning has been developed.

“Intelligent planning” [3], is the inference of the action, which is to achieve the preset goal as much as possible through the expected effect of the expected action, selection and organization of a series of actions. It is an interdisciplinary subject covering multiple fields such as knowledge expression, human-computer interaction, scenario calculation and knowledge reasoning [4-7].

In resource composition, the resource description is mapped to the formalized description of the action. In the resource space, the goal is to construct the resource composition, and the sequence of resource composition is obtained through formal reasoning to form the resource composition scheme dynamically.

Among many intelligent planning such as HTN planning, PDDL (Planning Domain Definition Language) and etc., we finally choose the HTN planning for the reason that the core idea of HTN planning is similar to the OWL-S process model as we have mentioned above, that is, resource composition is decomposed from one task to a specific resource.

![Figure 1. Part of a resource description file.](image)
The resource composition based on intelligent planning uses formal method or reasoning system. We should preprocess the resource description before execution, which is difficult for users. Also the complexity of the method is higher while the planning space grows. Also the classification of resources is still done manually now and usually divides resources into erroneous types.

As a result, we introduce the concept of resource type and use it as the planning space and propose the automatic resource classification model based on ontology and domain. Based on this model, the intelligent planning and adaptive technology are combined, and the adaptive strategy is dynamically selected on the basis of the results of the intelligent planning combination, so as to obtain better resource execution results.

3. Resource Clustering

3.1. Description of Resource Domain Ontology

We use OWL-S to define resource domain and resource type. We realize automatic classification of resource type instances through ontology.

Resource domain ontology describes a class of resources that have the same or similar functionality, so the resource domain ontology is suitable for the Class description in OWL-S, and it has no instance. However, resource type ontology describes resources with equivalent input and output parameters, so it defines the properties that the resource type should have, and it has instance.

3.2. The Improved Clustering Algorithm

Clustering refers to the classification of objects into close and independent clusters based on the information of the relationships between objects and their individuals, so that objects within clusters have the maximum similarity and objects between clusters have the minimum similarity. By clustering algorithm, the resource can be clustered according to the function and context.

K-means algorithm is the most classic algorithm. However it has some disadvantages such as the selection of K and initial cluster center. So we refer to the method proposed in [8]. The algorithm is as follows:

1. A point is randomly selected as the center point of the first class cluster;
2. The point farthest from the above point is selected as the center point of the second initial class cluster, and then the point with a larger distance from the first two points is selected as the center point of the third initial class cluster, and so on, until K initial class cluster centers are selected;
3. Then we use the K as the initial cluster of the K-means algorithm.

However there is no guarantee that the obtained K must be correct. Therefore, a measurement criterion is needed to determine whether the clustering has been completed correctly. We refer to the [9], the dispersion degree of a cluster can be expressed as in Equation (1):

$$\text{similarity}_i = \frac{\sum_{x \in c_i} \text{dist}(x, c_i)}{m_i}$$

(1)

where the C is the cluster, c is its center, dist is the distance of the object and the center, m is the size of cluster.

So we add one step after the above algorithm as follows:

4. We compute all the clusters. If the dispersion is greater than the given value, the cluster can be broken down into smaller clusters and update the K value, and the execution is continued back to step 3.

The clustering method can be defined as in Equation (2):

$$P = \{O, \text{dist}, SD, C\}$$

(2)

where O is the set of ontology, dist is the distance of them, SD is tightness of cluster, C is the set of clusters.
So our improved clustering algorithm is as shown in Fig. 2:

![Algorithm](image)

**Figure 2.** The improved clustering algorithm.

By clustering the functional resources, the resources in the system are divided into different resource domains according to their functional characteristics. At the same time, the resource context is clustered, and the resource can be performed under different environments to make the model adaptive.

For the reason of efficiency and the resource replace, we break the problem into two parts: task planning by HTN and resource allocation. Also, we have the resource register center.

4. Dynamic Resource Composition Model

4.1. Task Planning

Taking resource domain and resource type as planning object and using HTN planning to generate abstract resource composition sequence composed by resource domain or resource object, the scale of planning domain is controlled effectively and the efficiency of composition scheme generation is improved. HTN planning continues to break down the task into smaller ones until atomic tasks.

HTN planning module is divided into three parts as shown in Fig. 3: the problem converter describes the task problem according to the input and output parameter; the domain converter includes all resource type for operation; the planning engine is the most important part that it uses the former two files to break down the task into the resource type list into one Plan file and continue to the allocation step below [10].
4.2. Resource Allocation

MDLS algorithm is a quick heuristic algorithm for tasks and resources, that quickly organize and allocate existing resources according to the order of tasks, so as to guarantee the allocation of resources to complete tasks with high efficiency. Here we use the result in the Plan file above as the task in the MDLS algorithm [11].

Allocation of our resources is divided into two steps: the functional matching is to match resources based on the IOPE information (namely Input, Output, Precondition and Effect), and the QoS (time, cost and so on) matching is to compare multiple results with QoS to select the optimal resource instance. Then we can get the optimal resource instances and accomplish the task.

5. An Example

There is a military task that we may use computing and application resources to handle the issue. An organization X is going to make the decision immediately. There are 3 computing resources (compute1, compute2, compute3) under X and the resource information is in Table 1.

The application resources under X are the intelligence subscription service (subscribe), situational distribution (situationDistribution), threat assessment (threatAssessment), fusion of sea situation access (rrhqjrFun), fusion of air situation access (rrkqjrFun) five categories in Table 2.

| Resource  | Satisfaction | Cpu | Memory | Disk |
|-----------|--------------|-----|--------|------|
| compute1  | 50           | 2   | 6      | 256  |
| compute2  | 90           | 2   | 8      | 512  |
| compute3  | 80           | 1   | 8      | 512  |

| Resource                | Satisfaction |
|-------------------------|--------------|
| subscribe               | 50           |
| situationDistribution   | 90           |
| threatAssessment1       | 80           |
| threatAssessment2       | 90           |
| rrhqjrFun               | 50           |
| rrkqjrFun               | 90           |

First, we register these resources in our project, and each resource has its resource type, context domain and QoS domain, so we can easily manage them.
We can get the result in the Plan file as show in Fig. 4 by task planning step:

```plaintext
[Plan cost: 6.0
 (\compute)
 (\subscribe)
 (\situationDistribution)
 (\threatAssessment)
 (\rrhQijFun)
 (\rrkQijFun)

-------------
```

**Figure 4.** The Plan file.

The result of allocation step from our project shows that compute2 -> subscribe -> situationDistribution -> threatAssessment2 -> rrhQijFun -> rrkQijFun, that is, we can use these resource instances to execute this task. So our system verify the theory and method in the paper.

6. Conclusion
In order to plan resources dynamically in a short time according to the task requirement and allocate the optimal resource instance, we propose an adaptive dynamic resource composition model to realize fast searching and matching of resources through automatic classification of resources based on type, HTN planning and MDLS algorithm. We design a project and use an example to get the resource type list from the task and get the best resource for each resource type.

Acknowledgments
I would like to express my gratitude to Professor Tang Xinhui, my supervisor, for his encouragement and guidance during my writing of this paper and my classmates for their advice on my paper structure and format.

References
[1] Jiang Ge, Design and implementation of emergency task resource planning system based on conflict resolution, Shanghai Jiao Tong University, 2015.
[2] Y-S Tan, Vivekand Vellanki, Jie Xing, Brad Topol, and Gary Dudley, Service domains, IBM Systems Journal 43, 4 (2004), 734–755.
[3] Malik Ghallab, Dana Nau, and Paolo Traverso, Automated Planning: theory and practice, Elsevier, 2004.
[4] Jean-Charles Pomerol et al, Artificial intelligence and human decision making, European Journal of Operational Research 99, 1 (1997), 3–25.
[5] Blai, Bonet and H. Geffner, Planning and control in artificial intelligence: A unifying perspective, Applied Intelligence 14, 3 (2001), 237–252.
[6] Dana S Nau, Tsz-Chiu Au, Okhtay Ilghami, Ugur Kuter, J William Murdock, Dan Wu, and Fusun Yaman, SHOP2: An HTN planning system, Journal of artificial intelligence research 20 (2003), 379–404.
[7] David E Smith, Jeremy Frank, and Ari K Jónsson, Bridging the gap between planning and scheduling, The Knowledge Engineering Review 15, 1 (2000), 47–83.
[8] Tu zipei, Big data revolution, Guangxi Normal University Press, 2012.
[9] Wei-ning Qian and AY Zhou, Analyzing popular clustering algorithms from different viewpoints, Journal of software 13, 8 (2002), 1382-1394.

[10] Haiqing Wang, Peng Sun, Task Resource Planning and Verification Method Based on Intelligent Planning, 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), 2018.

[11] Wang JiangFeng, Research on operational task resource allocation algorithm based on MDLS and GA, National University of Defense Technology, 2005.