Journal of Industrial Information Integration 1 (2016) 20–25

Contents lists available at ScienceDirect

Journal of Industrial Information Integration

journal homepage: www.elsevier.com/locate/jii

Multi-criteria selection for services selection in service workflow

Wattana Viriyasitavat

Business Information Technology Division, Department of Statistics, Chulalongkorn University, Pathumwan, Bangkok 10330, Thailand

ARTICLE INFO

Article history:
Available online 4 March 2016

Keywords:
Service
Workflow
Compliance checking
Specification language
MCDM
AHP
Algorithm

ABSTRACT

Selecting services to be part of a workflow has been a very important challenge. As a number of available services increases, the selection becomes more complicated. Different providers may offer the same service with different non-functional attributes such as services' qualities, past experience, reputation value, etc. Moreover, the importance of each attribute is subjective and varies in different contexts of use. Complexity increases due to dynamic changes in real-time service workflow interoperation, for example, services can dynamically join or leave at any time, attributes can be changed, or the importance of an attribute can be lessened or increased. To alleviate this problem, this paper presents a Multi-Criteria Decision Making (MCDM) approach for dynamic real-time service selection in service workflow. The main study emphasizes on an integrated architecture with the enhancement of compliance checking for Service Workflow Specification language (SWSpec) with MCDM using Analytic Hierarchy Process (AHP). This approach enables real-time service selection based on the degree of compliance, in which depending on each context the best-suited services can be determined. To make this approach more understandable, an application example of car rental agent is demonstrated.

1. Introduction

Recently, service based interactions in the form of workflows have become popular [27]. They are utilized in several areas such as e-commerce, e-business, e-organizations, Grid, Virtual Organization, and Cloud Computing [24]. Services are utilized as building blocks to execute tasks as a part of business processes such as order, payment, billing, stock management, and many more.

One of the important challenges is how to select suitable services to be part of a workflow. A service workflow typically consists of several tasks where each task can be executed by one or many (composite) service(s). During runtime, services can be removed o replaced if some of their attributes are no longer satisfied by pre-defined requirements. This leads to more complicated service selection problem than the previous traditional composition approaches [2,5,9,10,31,32].

In opened environment, typical service selection is subjected to various, qualitative or quantitative, attributes including service qualities, reputation value, past experience, and many more. This is because one service can be offered from several providers with different attributes. A set of required attributes is subjected to a particular context or situation. Nowadays, requirement specification language is used as a tool to formally and mathematically encapsulates requirements, defining attributes required for a service to execute a task [7,8,16,25]. The degree of compliance of such requirements reflects trust of a service(s). Therefore, a service with higher compliance value is likely to be selected.

In the area of service workflow, SWSpec is a formal requirement specification language specifically developed to specify services' properties [25]. Requirements are encapsulated into formal expressions, enabling real-time automatic compliance checking [26,29,30,31]. However, one major limitation is that the compliance checking algorithms generate binary answers, true (completely satisfied) or false (completely unsatisfied). The false implies a service is unsatisfied which will not be selected to a workflow and vice versa. This solution is impractical in the situation when some requirements are not designed to have precise true or false, and the importance of requirements can vary.

To alleviate the problem, this paper presents an integrated approach enhancing SWSpec compliance checking algorithms with AHP calculation. The main purpose of using AHP is to rank the relative importance for each SWSpec requirement and then evaluate compliance value. The solution provides an additional dimension of decision-making, enabling better service selection in service workflow.

2. Related works

AHP is a popular MCDM tool that has been used widely in almost all decision-making related problems including, planning, selecting a best alternative, resource allocations, optimization, etc.

E-mail address: wattana@cbs.chula.ac.th

http://dx.doi.org/10.1016/j.jii.2016.03.003
S2452-414X(16)00005-4/

[23]. One of its capabilities is the evaluation of relative ranking among available alternatives. Criteria are first defined in the form of a hierarchical tree, and information regarding the importance of each criterion is then evaluated using pair-wise comparisons [20]. For comparison, the values of both the quantitative and qualitative criteria are normalized into a numeric scale, ranging from 1 to 9. This section summarizes AHP related works in the 'selecting a best alternative' and 'resource allocations', which are closely related to our approach.

In the area of selection, Lai et al. [14,15] and Jung and Choi [11] explained the applications of AHP-based MCDM for selecting software. The hierarchy for pair-wise comparison was formed with criteria including properties such as interface, multimedia, and cost effectiveness. Kengpol and O'Brien [13] demonstrate the use of AHP for the selection of advanced technology focusing on cost-benefit and effectiveness criteria. Al Harbi [1] and Tam and Tummala [21] applied AHP to select the best vendor with criteria such as experience, financial stability, and quality performance. In resource allocation, Andijani and Anwarul [3] proposed an AHP-based framework to identify the best allocation of buffer in serial production system.

In the aspect of resource allocation, Ramanathan and Ganesh [19] presented a general framework for resource allocation, where AHP is mainly exercised to determine coefficients of linear programming functions. Recent work on AHP-based service allocation has been proposed by Rehman et al. [22]. They developed a multi-criteria methodology for selecting cloud services according to end user requirements. The intensive review of AHP applications, presented by Vaidya and Kumar [23], evidences that AHP is largely accepted as an efficient tool for decision-making.

3. Backgrounds

3.1. Service Workflow Specification (SWSpec) language

SWSpec language is invented to mathematically encapsulate and specify requirements in service workflows [25], from both the perspectives of a workflow owner and participating services. It works successfully to capture the structural characteristic of a service workflow that consists of several tasks, where the coordination of tasks forms paths and branches of a workflow. Fig. 1 illustrates an example of service workflow where shaded rectangle represents tasks and circle represents services. Each task can be executed by one or composite services. For example, three services with attributes *p*, *q*, and *r* can be part of task 1 execution. A workflow owner usually enforces service selection requirements as decision metric through SWSpec language that indicates properties of services needed for a particular task. In the perspective of services, they can employ SWSpec to express their requirements to specify properties of other services collaborated in the same workflow.

The formal syntax of SWSpec to forms formulae is defined in three classes. (1) Composite formulae specify property composition of services for a task, (2) Path formulae quantify paths and branches of a workflow, and (3) Direction formulae indicate the direction to which the formulae applied. SWSpec grammar is shown below.

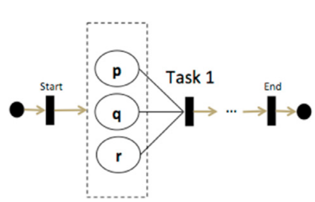

Fig. 1. Example of required attributes *p*, *q*, and *r* for task 1 execution.

1. Composite formulae

$$S ::= !S \mid \mathcal{F}E_tZ \mid \mathcal{P}E_tZ \mid \mathcal{F}A_tZ \mid \mathcal{P}A_tZ \mid (S\&S) \mid (S \mid\mid S) \text{(Quantifier part)}$$
$$Z ::= \varepsilon \mid !Z \mid (Z \sqcap Z) \mid (Z \sqcup Z) \mid (Z \boxplus Z) \text{(Property part)}$$

2. Path formulae

$$R ::= \top \mid \bot \mid S \mid \sim R \mid (R \wedge R) \mid (R \vee R) \mid \exists_t \odot R \mid \exists_t \diamond R \mid \exists_t \Box R \mid \exists_t [R \Cup R] \mid \exists_t [R \text{ U. } R] \mid \forall_t \odot R \mid \forall_t \diamond R \mid \forall_t \Box R \mid \forall_t [R \Cup R] \mid \forall_t [R \text{ U } R]$$

3. Direction formulae

$$W ::= \top \mid \bot \mid \mathcal{H}R \mid \mathcal{B}R \mid \sim W \mid (W \wedge W) \mid (W \vee W)$$

Each SWSpec requirement uses an inductive definition of Composite, Path and Direction formulae. Composite formulae are specially invented to specify attributes of services to a task (t) (subscripted in E_t and A_t). This is the place where requirements for service selection actually take actions and AHP can be applied from this point. Below, the definitions of each Composite operator are explained. Due to space limitations, the complete details of Path and Direction formulae are referred to [25].

Composite formulae are composed of two parts.

(1) The quantifier part consists of Forward ($\mathcal{F}$) and Previous ($\mathcal{P}$) operators, indicating target services from the viewpoint of task (t). $\mathcal{F}$ targets at services after a task (services requires the result of a task execution). $\mathcal{P}$ targets services selected (composed) for a task (t). Composite For Some (E_t) and Composite For All (A_t) quantify services indicated by $\mathcal{F}$ or $\mathcal{P}$. E_t requires at least one service, while A_t requires all services, to satisfy property *Z*. The remaining operators correspond to the definitions in first order logic.

(2) The property part defined properties (*Z*) of services required for a task (t). Composite Conjunction ($Z_1 \sqcap Z_2$) indicates services composed for a task must contain both properties Z_1 and Z_2. Composite Disjunction ($Z_1 \sqcup Z_2$) indicates services composed for a task (t) must contain either Z_1 or Z_2. Lastly, Composite Exclusive Disjunction ($Z_1 \boxplus Z_2$) indicates that services composted for a task must contain one and only one of Z_1 or Z_2. The property part must be preceded by one of the quantifier operators. For example, Previous operator ($\mathcal{P}$) with Composite For Some (E_t) and Composite Conjunction ($\sqcap$) is expressed as $\mathcal{P}E_t(Z_1 \sqcap Z_2)$.

3.2. Analytical Hierarchy Process (AHP)

Several AHP applications have been developed by various researchers and have been proven useful for prioritizing alternatives among multi-criteria and multi-attributes [4,6,12,17,18]. A typical AHP use a definite number 1–9 scale for the pairwise comparison, ranging from 1 (weakly important) to 9 (absolutely more important). The corresponding reciprocals 1, 1/2, 1/3, ..., 1/9 are used for the reverse comparison. The AHP method is outlined in the following steps.

Step 1: Experts on the basis of their knowledge are required to determine the ranking of each factor. A precise comparative numerical value is provided. For example, let F_1, F_2 and F_3, be reliability, availability, and reputation factors respectively, reliability is three times as important as availability, while it is nine times as reputation. The scale is ranging from 1 to 9.

Step 2: Let $F_1, F_2, \ldots, F_n$ the factors, pair-wise comparison between F_i and F_j representing the quantified judgment is expressed