P-Spec Policy Model-based Service Selection Approach for Privacy-aware Service Composition Establishment

Yun-fei MENG¹, Zhi-qiu HUANG¹ and Chang-bo KE²

¹Nanjing University of Aeronautics and Astronautics, Nanjing, China
²Nanjing University of Posts and Telecommunications, Nanjing, China

Keywords: Service selection, P-Spec model, Privacy-aware service, Cloud computing.

Abstract. As the issue of privacy exposure is becoming more and more serious, many scholars put forward the concept of privacy-aware service composition. Inspired by this concept, we propose a P-Spec policy model-based service selection approach for constructing privacy-aware service composition. P-Spec proposed in this paper is a kind of formal policy specification language, which can be utilized to describe the heterogeneous privacy policies and privacy preference explicitly. With the relevant P-Spec models, we further conceive the framework of our approach, which consists of a group of P-Spec policy models and a specific policy matching algorithm. To verify the feasibility and effectiveness of our approach, we implement a policy matching engine prototype and evaluate its performance.

Introduction

In cloud computing era, service consumers often need to disclose personal information to service providers, such as name, age, occupation, locations et al., during their utilizing the variety of Web applications. Once personal information is disclosed online, it may be abused or accessed without authorization, thus increasing the risk of illegal collection, usage and disclosure of personal sensitive data [3–5]. As the issue of privacy exposure is becoming more and more serious, many scholars put forward the concept of privacy-aware service composition which is to limit the propagating scope of consumer’s personal privacy information through a strict protective mechanism [8–10]. In their opinions, not only the functional requirements, the privacy-aware services also need to satisfy those nonfunctional constraints, such as the privacy constraints. In this paper, we conceive that a privacy-aware service composition (PASC) should be a set composed by those services which their privacy policies can well satisfy the consumer’s privacy requirements or preferences. And any service within PASC can access the consumer’s privacy data freely, while the services outside of PASC must have the relevant authorization from consumers if they want to access the consumer’s privacy data.

The contributions of this paper can be concluded as follows. Firstly, we propose a kind of formal policy specification language P-Spec which can be utilized to describe the variety of heterogeneous privacy policies at using explicitly. Secondly, we propose a service selection approach with P-Spec models to select out a group of privacy optimal services, i.e. privacy-aware service composition. The framework of our approach consists of a group of P-Spec policy models and a specific policy matching algorithm. Thirdly, to verify the efficiency and feasibility of our approach, we implement a proof-of-concept policy matching engine prototype and evaluate its performance in terms of entering the different scale of policy models.

P-Spec Policy Specification Language

Definition 1 (Grammar of P-Spec language): The context-free grammar of P-Spec is a 4-tuple: $G^{\text{P-Spec}} = <N, T, R, S>$, where $N$ is the finite set of non terminal symbols. $T$ is the finite set of terminal symbols. Let $T = \{OP, C, K, V\}$, where $OP = /, =, >=, <=, ! =, \&\&$, $||, \rightarrow, (), /$ is the finite set of operation symbols, semantically, $=$ represents equal, $\geq$ represents bigger than or equal, $\leq$ represents less than or equal, $! =$ represents not equal, $\&\&$ represents logical and, $||$ represents logical or, and $\rightarrow$
represents logical imply to. $C$ is the finite set of constant symbols. $K$ is the finite set of key word symbols. $V$ is the finite set of variable symbols. $R$ is the finite set of production rules, each production rule of $R$ is a relation from $N$ to $(N \cup T)^*$, where the asterisk represents the kleene star operation. Given $\phi \in N$, $c \in C$, $k \in K$, $v \in V$, $R$ can be defined with Backus-Naur Forms as follows. $\phi ::= k | k = v | v = c | k >= v | v >= c | k <= v | v <= c | k != v | v != c | (\phi) | \phi || \phi | \phi \& \& \phi | \phi \rightarrow \phi$. And $S$ is the start symbol of P-Spec language.

**Definition 2 (P-Spec Language):** Given grammar $G^{P\text{-Spec}} = <N, T, R, S>$, the formal language P-Spec can be defined as a set: $L(G^{P\text{-Spec}}) = \{\omega \in T^* | S \Rightarrow \omega\}$, where $T^*$ is the universal set of strings produced by $T$. $S \Rightarrow \omega$ represents from the start symbol $S$, the string $\omega$ can be derived with relevant production rules of $R$ within limited steps.

**Definition 3 (Interpretation):** An interpretation of $L(G^{P\text{-Spec}})$ is a pair $<A, I>$, where $A$ is nonempty set and $I$ is a function which maps the $N$, $T$, $R$, $S$ of $L(G^{P\text{-Spec}})$ into the $I(N)$, $I(T)$, $I(R)$, $I(S)$ of $A$.

**Definition 4 (Model of P-Spec language):** Given an interpretation of $<A, I>$, if $<A, I>$ has mapped the $N$, $T$, $R$, $S$ of P-Spec language into the $I(N)$, $I(T)$, $I(R)$, $I(S)$ of $A$, then $M^{P\text{-Spec}} = <A, I(N), I(T), I(R), I(S)>$ is a model of P-Spec language, and $A$ is called the universe (or domain) of model $M^{P\text{-Spec}}$.

**The Framework of Our Approach**

The framework of our approach can be depicted as Figure 1.

![Figure 1. The framework of our approach.](image)

In our approach, we set up a policy matching engine (PME) deployed in a middleware between service consumers and service providers to execute policy matching and service selection. Firstly, the heterogeneous privacy policies of different service providers, such as nature language based or XML based policies, are formally specified as a unified service privacy policy model (SPPM), and meanwhile, the heterogeneous privacy preferences of different consumers are formally specified as a unified consumer privacy preference model (CPPM) correspondingly. The stage of formal specification is very crucial for our approach because the semantic consistency of expressions and symbols is the fundamental of execution of any policy matching. After formal specifications, $R_S$ rules of SPPM and $R_{C_j}$ rules of CPPM will be delivered into the policy matching algorithm (PMA) and to be executed automatically. After multi-round policy matching, the privacy policy matching model (PPMM) which is exactly composed by a set of policy matching $R_{M_k}$ rules will be produced. With PPMM, any service consumer of $C_i$ can retrieve a group of privacy optimal services (PASC) using the key word of $C_i$, and PASC is the privacy-aware service composition for consumer of $C_i$.

**Consumer Privacy Preference Model (CPPM)**

**Definition 5 (Consumer Privacy Preference Rule):** The privacy preference rule model is an interpretation for P-Spec language under the universe about consumer privacy preferences. Let the interpretation as follows. $I(N) = \{RC_i, Preference\}$, with $0 \leq i \leq n$, $i \in N$. $I(K) = \{C, Target, Role, SP, Retention, Item\}$. $I(V) = \{C_i, T_i, RO_i, SP_i, TR_i, X_i\}$, with $0 \leq i \leq n$, $i \in N$. $I(R)$ is defined as follows.
Definition 12 (Privacy Level): Given a $RC_i \in CPPM$ and a $RC_j \in CPPM$, if there exists:

1. $T_i \in RS_n \not\exists T_j \in RC_j$ and $T_i = T_j$;
2. $RO_i \in RS_n \not\exists RO_j \in RC_j$ and $RO_i = RO_j$;
3. $SP_i \in RS_n \not\exists SP_j \in RC_j$ and $SP_i = SP_j$;
4. $X_i \in RS_n \not\exists X_j \in RC_j$ and $X_i = X_j$;
5. $TR_i \in RS_n \not\exists TR_j \in RC_j$ and $TR_i \leq TR_j$;
then, the SP rules complies with the $RC_i$, we denote it as: $RS_i \prec RC_j$.

Definition 9 (Policy Matching Rule): Given a $RS_i \in SPPM$, let $RS_i := (SP = SP_i) \rightarrow Policy$. Given a $RC_j \in CPPM$, let $RC_j := (C = C_j) \rightarrow Preference$. if $RS_i \prec RC_j$, then $RS_i$ and $RC_j$ can be merged into a privacy policy matching rule of $RM_k$. Let $RM_k := (C = C_j) \& \& (SP = SP_i) \rightarrow Policy$.

Definition 12 (Privacy Level): Given a $RS_i \in SPPM$, $\forall X_i \in RS_i$, we assign an integer $\lambda_{Xi} \in [0, 10]$ to $X_i$, and let:

$$RC_i := (C = C_i) \rightarrow Preference, with Preference := (Target = T_i) \& \& (Role = RO_i) \& \& (SP = SP_i) \& \& (Retention \leq TR_i) \& \& (Item = X_i) \& \& Preference \& \& Preference \& \& Preference \& \& Preference \& \& Preference.$$
\[
\Omega = \sum_{i=0}^{n} \lambda x_i
\]

(1)

Then \( \Omega \) is the privacy level of this \( RS_i \) rule.

Definition 13 (Depth): Given a \( RS_i \in SPPM \), \( I(V) \) is the set of all variables of this \( RS_i \) rule, we let:

\[
\Delta = ||I(V)||
\]

(2)

Then \( \Delta \) is the depth of the \( RS_i \) rule.

Definition 14 (Privacy-aware Service Composition): Given a consumer of \( C_i \) and PPMM, for \( \forall RM_j \in PPMM \), if \( \exists RM_j \in PPMM, C_j \in RM_j \) and \( C_j = C_i \), then push all of \( SP_j \in RM_j \) into the set of \( PASC \), thus \( PASC \) is the privacy-aware service composition for consumer of \( C_i \).

Based on these pre-defined definitions, we can conclude our policy matching algorithm as a 3-round matching procedure.

1st Round: Given a \( RC_j \) rule of CPPM, we seek out all of \( RS_i \) rules which can \( RS_i \preceq RC_j \). If we can seek out the best \( RS_i \) rule from SPPM, then we reach a policy matching rule of \( RM_k \) and stop. Otherwise we start the 2nd round matching.

2nd Round: If no any \( RS_i \) rules can comply with the \( RC_j \) rule, then we calculate the privacy level \( \Omega \) of each \( RS_i \) rule in SPPM, and return the \( RS_i \) rule which has the minimum privacy level. If more than one \( RS_i \) rules can comply with the \( RC_j \) rule, then we calculate the privacy level \( \Omega \) of each \( RS_i \) rule having found, and return the \( RS_i \) rule which has the minimum privacy level. If we can seek out the best \( RS_i \) rule, then we reach a policy matching rule of \( RM_k \) and stop. Otherwise we start the 3rd round matching.

3rd Round: If more than one \( RS_i \) rules have the same privacy level, then we calculate the depth \( \Delta \) of each \( RS_i \) rules having found, and return the \( RS_i \) rule which has the minimum depth value and reach a \( RM_k \) rule.

Implementation and Evaluation

To verify the feasibility of our approach, we implement a C++ version proof-of-concept policy matching engine (PME) prototype on Windows platform to simulate the real privacy policy matching in cloud computing environment. All these experiments are performed with Open-source Qt Creator 3.5.1 running on Windows-7 32bit for a machine of Intel (R) Core(TM) i5 CPU @ 2.50GHz, 4.0G of RAM.

Firstly, according to the data set of privacy policies and preferences originated from the reference [1], we can construct the CPPM and SPPM using PME prototype. After policy matching, the first, second and third round policy matching results can be shown as Figure 3.

![Figure 3. 1st round result (left), 2nd round result (mid), 3rd round result (right).](image)

Furthermore, in order to evaluate the execution performance of our PME in terms of entering the different scale of SPPM and CPPM, we design two experiments for comparative analysis. The experiment 1 is to limit the input number of \( RS_i \) rules of SPPM is at 10, and we increase the input number of \( RC_j \) rules of CPPM from 10 to 100 continuously. The increased \( RC_j \) rules are composed together randomly, and no any special arrangement. And then, we record the corresponding time overhead (in microsecond) of executing the 1st round policy matching each time and produce a testing
result of experiment 1. The experiment 2 is to limit the input number of $RC_i$ rules of CPPM is at 10, and we increase the input number of $RS_j$ rules of SPPM from 10 to 100 continuously. Under the same reason with experiment 1, the increased $RS_j$ rules are composed together randomly, and no any special arrangement. We record the time overhead (in microsecond) of 1st round policy matching each time and produce a testing result of experiment 2. Having executed the two experiments respectively, we push the record of the two experiments together and create a unified performance evaluation diagram which is shown in Figure 4.

![Figure 4. Performance evaluation of 1st round policy matching.](image)

**Summary**

We proposed a P-Spec model-based service selection approach for constructing the privacy-aware service composition. The main idea of our approach is to formalize the heterogeneous privacy policies and privacy preferences as the unified SPPM and CPPM with P-Spec policy specification language. Based on the constructed P-Spec policy models, we can seek out a group of $RS_i$ rules which can comply with the $RC_i$ rules of given consumer, and the privacy-aware service composition can be derived from these matched $RS_i$ rules definitely. We implemented a policy matching engine prototype to verify our approach, and the experimental results illustrate that our approach is feasible and effective. In the future, we have planed to develop a testing policy matching engine which can be deployed into the real cloud computing environment, and conduct some experiments to further evaluate the feasibility of our approach under the multi-users and high-concurrency networking environment.

**Acknowledgement**

This research was financially supported by the National Natural Science Foundation of China (Grant 61602262), Jiangsu Natural Science Foundation of China (Grant BK20150865), Jiangsu University Natural Science Foundation (Grant 15KJD520001), NUPTSF (Grant No. NY214164, NY215120), China Postdoctoral Science Foundation Funded Project (Project No. 2016M591842) and Jiangsu Province Postdoctoral Science Foundation Funded Project (Project No. 1601198C).

**References**


